

GAME TRAFFIC ANALYSIS AND SIMULATION IN FIRST PERSON SHOOTER ENVIRONMENT

QILI ZHOU

**A thesis submitted in partial fulfilment of the
requirements of the University of Abertay Dundee for
the degree of Doctor of Philosophy**

**University of Abertay Dundee
United Kingdom**

June 2008

ABSTRACT

Internet delay is caused by transmission medium and also by routing and queuing. Although delay is often tolerable in many applications such as file transfer or Web, it does affect performances for delay-sensitive applications such as IP phone and video streaming. For First Person Shooter (FPS) games, where delay is even more important, Internet delay can determine who wins or loses a game. The previous research showed that player at a distance was disadvantaged due to larger delay. Network traffic for such games was analysed using packet size and inter-packet time metrics fitting a statistical model. However, no research explored the relation between “clean” and “interfered” traffic, only Joyce (2000), Bangun (2000), Jehaes (2003) and Carrig (2005) explored the interactions between game and other traffic, and the effect of other traffic on the game server-client delay was overlooked. This dissertation explores the traffic generated by an FPS game, using Quake III as an example, taking into account the effects of network delays.

An FPS game was played both in an isolated local network with no other traffic and was repeated across the Internet to explore the difference between “clean” test-bed traffic and realistic Internet “interfered” traffic. The metrics of packet size and inter-packet time of empirical traffic were analysed. A simple statistical approach was applied using goodness-of-fit test based on the probability calculation to produce a mathematical model. The results were used to set up a simulator experiment to further explore the behaviour of game traffic under varying degrees and also different types of network traffic.

The approximate statistical distributions for “clean” test-bed game traffic were based on packet size and inter-packet time. As to packet-size, server traffic had extreme probability distribution, while client traffic was simply distributed in a short interval. As to inter-packet time, both server and client had deterministic distribution. Inter-arrival time in “interfered” network was totally different from inter-send time and was network dependent, and this was confirmed in a simulation experiment. The difference with results in previous research was also explained by the simulation results. The game delay was not only affected by volume but also by inter-packet time behaviour of network traffic, because simulation showed that traffic streams with same parameters but different inter-packet time affected game traffic delay differently. The pseudo game traffic generated by analytical distribution can replace the realistic game traffic in simulation experiment.

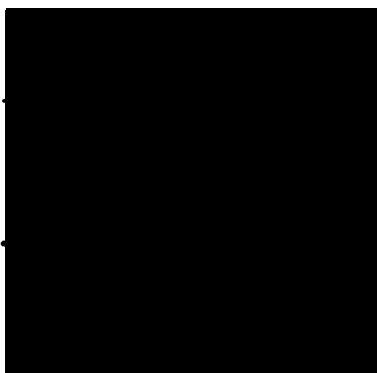
ACKNOWLEDGEMENTS

I would like to thank Colin Miller, Victor Bassilious, Harry Staines and Peter Astheimer for their help.

DECLARATION

I declare that the work in this dissertation is original and was performed by the author.

Signed ..



Date 13/8/2008

6/10/'08

TABLE OF CONTENTS

CHAPTER 1	LITERATURE SURVEY	1
1.1	INTRODUCTION	1
1.1.1	COMPLEXITY OF INTERNET TRAFFIC	1
1.1.2	DELAY-SENSITIVE APPLICATIONS	2
1.1.3	GAME TRAFFIC AND SERVER HOSTING	3
1.1.4	FIRST PERSON SHOOTER GAMES	3
1.2	DISCREPANCIES IN PUBLISHED RESULTS	4
1.2.1	DISCREPANCIES IN PUBLISHED RESULTS FOR THE GAME QUAKE	5
1.2.2	DISCREPANCIES IN PUBLISHED RESULTS FOR THE GAME COUNTER- STRIKE	6
1.2.3	DISCUSSION ON CAUSES FOR DISCREPANCIES	7
1.2.4	INTER-SEND VERSUS INTER-ARRIVAL TIME	8
1.2.5	"CLEAN" VERSUS "INTERFERED" TRAFFIC	9
1.2.6	OBJECTIVE ONE	10
1.3	IMPACT ON GAME TRAFFIC FROM OTHER TRAFFIC	10
1.3.1	INTERACTIONS BETWEEN GAME TRAFFIC AND OTHER TRAFFIC	10
1.3.1.1	TCP AND UDP PROTOCOLS	10
1.3.1.2	INTERACTIONS BETWEEN GAME TRAFFIC AND OTHER TRAFFIC	11
1.3.1.3	GAME PACKET DELAY AND INTER-ARRIVAL TIME	12
1.3.2	ROLE OF OTHER TRAFFIC IN THE CHANGE OF GAME TRAFFIC	13
1.3.3	OBJECTIVE TWO	14
1.4	AN ANALYTICAL TRAFFIC MODEL	14
1.4.1	OBJECTIVE THREE	15
1.5	GAME SELECTION	15
1.5.1	DOUBTS ON CLASSIC WORK	15
1.5.2	QUAKE III ENGINE AND QUAKE III ARENA	16
1.5.2.1	QUAKE III ENGINE	17
1.5.2.2	SELECTION OF QUAKE III ARENA	18
1.6	OBJECTIVES OF THIS RESEARCH	18
CHAPTER 2	METHODS	19
2.1	OVERALL PLAN OF RESEARCH	19
2.2	TEST-BED EXPERIMENT	20
2.2.1	DESIGN OF TEST-BED EXPERIMENT	20
2.2.1.1	AN ISOLATED TEST-BED	20
2.2.1.2	GAME PARTICIPANTS, DURATION AND SESSIONS	21
2.2.1.3	PACKET SNIFFER	21
2.2.2	DATA COLLECTION OF TEST-BED EXPERIMENT	22
2.2.2.1	PARSING ETHERREAL FILE	22
2.2.2.2	DELETING SOME DATA	23
2.3	DATA ANALYSIS OF TEST-BED EXPERIMENT	23
2.3.1	GOODNESS-OF-FIT TEST	24
2.3.1.1	CHI-SQUARE TEST	24
2.3.1.2	KOLMOGOROV-SMIRNOV TEST	25
2.3.2	CURVE FITTING PROGRAM	26
2.3.2.1	MATLAB	26

2.3.2.2	GNUPLOT	27
2.3.3	AN APPROXIMATE METHOD OF CURVE FITTING	27
2.3.3.1	SACRIFICE IN PREVIOUS RESEARCH	27
2.3.3.2	SELECTION OF KOLMOGOROV-SMIRNOV TEST	28
2.3.3.3	M-TRIAL TEST	29
2.3.3.4	PROCEDURE OF DATA ANALYSIS	30
2.4	INTERNET EXPERIMENT	30
2.4.1	DESIGN OF INTERNET EXPERIMENT	30
2.4.1.1	SERVER SELECTION	31
2.4.1.2	CLIENT TRAFFIC	31
2.4.2	DATA COLLECTION AND ANALYSIS	32
2.5	SIMULATOR EXPERIMENT	32
2.5.1	INTRODUCTION TO SIMULATION	33
2.5.1.1	NETWORK SIMULATION	33
2.5.1.2	NETWORK SIMULATOR	34
2.5.2	PROGRESSION FROM “CLEAN” TO “INTERFERED” TRAFFIC	34
2.5.2.1	NETWORK CONFIGURATION	35
2.5.2.2	GAME TRAFFIC BY ORIGINAL TRACE	36
2.5.2.3	OTHER TRAFFIC GENERATION	37
2.5.2.4	RESULT INTERPRETATION	38
2.5.2.5	PROGRAMMING TOOLS	39
2.5.2.6	PROGRESSION EXPERIMENT	39
2.5.3	EFFECTS FROM DIFFERENT TYPES OF TRAFFIC	40
2.5.3.1	METRICS	40
2.5.3.2	OTHER TRAFFIC	41
2.5.3.3	DATA COLLECTION	41
2.5.3.4	DATA ANALYSIS AND MANN-WHITNEY TEST	42
2.5.4	ANALYTICAL TRAFFIC MODEL	43
CHAPTER 3	RESULTS	44
3.1	RESULTS OF TEST-BED EXPERIMENT	44
3.1.1	INTER-SEND VERSUS INTER-ARRIVAL TIME	44
3.1.2	CLIENT INTER-SEND TIME	46
3.1.2.1	CLIENT INTER-SEND TIME OVERVIEW	46
3.1.2.2	DETERMINISTIC DISTRIBUTION	47
3.1.2.3	SUPPLEMENT	47
3.1.3	SERVER INTER-SEND TIME	48
3.1.3.1	SERVER INTER-SEND TIME PLOTS	48
3.1.3.2	SERVER INTER-SEND TIME DATA	49
3.1.4	CLIENT PACKET SIZE	50
3.1.5	SERVER PACKET SIZE	51
3.1.5.1	SERVER PACKET SIZE OVERVIEW	51
3.1.5.2	EXTREME DISTRIBUTION	53
3.1.5.3	PARAMETERS	54
3.1.5.4	CURVE FITTING	55
3.1.6	BASIC STATISTICS	57
3.2	RESULTS OF INTERNET EXPERIMENT	57
3.2.1	REGULAR AND IRREGULAR PLOTS	58
3.2.2	INTER-PACKET TIME	58
3.2.2.1	CLIENT INTER-SEND TIME	58
3.2.2.2	SERVER INTER-ARRIVAL TIME	60
3.2.3	CLIENT PACKET SIZE	67

3.2.4	SERVER PACKET SIZE	67
3.3	RESULTS OF SIMULATOR EXPERIMENT	68
3.3.1	PROGRESSION FROM “CLEAN” TO “INTERFERED” TRAFFIC	69
3.3.2	EFFECTS FROM DIFFERENT TRAFFIC	72
3.3.2.1	SERVER INTER-ARRIVAL TIME	74
3.3.2.2	SERVER-TO-CLIENT DELAY	75
3.4	EVALUATION OF ANALYTICAL TRAFFIC MODEL	76
3.4.1	APPROXIMATE AND FINE TUNED MODELS	76
3.4.1.1	FINE TUNING OF ANALYTICAL MODEL	77
3.4.1.2	ANALYTICAL DISTRIBUTION PER-CLIENT	77
3.4.1.3	ANALYTICAL GAME TRAFFIC MODEL	78
3.4.2	EVALUATION OF TRAFFIC METRICS	79
3.4.2.1	CLIENT PACKET SIZE	81
3.4.2.2	CLIENT INTER-SEND TIME	81
3.4.2.3	CLIENT INTER-ARRIVAL TIME	82
3.4.2.4	SERVER PACKET SIZE	83
3.4.2.5	SERVER INTER-SEND TIME	84
3.4.2.6	SERVER INTER-ARRIVAL TIME	85
CHAPTER 4	DISCUSSIONS, CONCLUSIONS AND FUTURE WORK	86
4.1	DISCUSSIONS	86
4.1.1	UNDERSTANDING SERVER INTER-SEND TIME	86
4.1.2	UNDERSTANDING BASIC STATISTICS	87
4.1.3	INTER-PACKET TIME AND QUEUING DELAY	88
4.1.4	CAUSES TO THE DISCREPANCIES	90
4.2	CONCLUSIONS AND FUTURE DIRECTIONS	92
4.2.1	CONCLUSIONS	92
4.2.1.1	GAME TRAFFIC CHARACTERISTICS	92
4.2.1.2	EFFECTS ON GAME FROM OTHER TRAFFIC	94
4.2.1.3	APPLICABILITY OF ANALYTICAL MODEL	96
4.2.2	FUTURE WORK AND POSSIBLE DIRECTIONS	97
4.2.2.1	MODELLING METHOD	97
4.2.2.2	INTER-ARRIVAL TIME	97
4.2.2.3	DELAY WITH ROUTING TECHNIQUE	98
4.2.2.4	DELAY WITH SERVICE POLICY	98
4.2.2.5	OTHER WORK	99
REFERENCES		100
APPENDIX A	CALCULATION OF M-TRIAL TEST	103
APPENDIX B	PARETO DISTRIBUTION	105
APPENDIX C	COMPOSITION OF INTERNET DELAY	106
APPENDIX D	PUBLISHED PAPERS	108

LIST OF FIGURES

FIGURE 1	COMMUNICATION AMONG HOSTS, CLIENTS AND SNIFFERS	20
FIGURE 2	NETWORK CONFIGURATION OF SIMULATION	36
FIGURE 3	SERVER INTER-SEND TIME IN TEST-BED EXPERIMENT	45
FIGURE 4	SERVER INTER-ARRIVAL TIME IN TEST-BED EXPERIMENT	45
FIGURE 5	CLIENT INTER-SEND TIME (ALL CLIENTS)	46
FIGURE 6	CLIENT INTER-SEND TIME (AMPLIFIED)	48
FIGURE 7	SERVER INTER-SEND TIME TO ALL CLIENTS	49
FIGURE 8	CLIENT PACKET SIZE DISTRIBUTION	51
FIGURE 9	SERVER PACKET SIZE IN TEST BEST EXPERIMENT, SESSION 1 (TO 4 CLIENTS)	52
FIGURE 10	SERVER PACKET SIZE IN TEST BEST EXPERIMENT, SESSION 2 (TO 4 CLIENTS)	52
FIGURE 11	SERVER PACKET SIZE IN TEST BEST EXPERIMENT, SESSION 3 (TO 5 CLIENTS)	53
FIGURE 12	CLIENT INTER-SEND TIME IN TEST-BED EXPERIMENT	59
FIGURE 13	CLIENT INTER-SEND TIME IN INTERNET EXPERIMENT, SESSION 1	59
FIGURE 14	CLIENT INTER-SEND TIME IN INTERNET EXPERIMENT, SESSION 2	60
FIGURE 15	SERVER INTER-ARRIVAL TIME IN TEST-BED EXPERIMENT	61
FIGURE 16	SERVER INTER-ARRIVAL TIME IN INTERNET EXPERIMENT, SESSION 1	62
FIGURE 17	SERVER INTER-ARRIVAL TIME IN INTERNET EXPERIMENT, SESSION 2	62
FIGURE 18	SERVER INTER-ARRIVAL TIME IN INTERNET EXPERIMENT, SESSION 3	63
FIGURE 19	SERVER INTER-ARRIVAL TIME IN INTERNET EXPERIMENT, SESSION 4	63
FIGURE 20	SERVER INTER-ARRIVAL TIME IN INTERNET EXPERIMENT, SESSION 5	64
FIGURE 21	SERVER INTER-ARRIVAL TIME IN INTERNET EXPERIMENT, SESSION 6	64
FIGURE 22	SERVER INTER-ARRIVAL TIME IN INTERNET EXPERIMENT, SESSION 7	65
FIGURE 23	SERVER INTER-ARRIVAL TIME IN INTERNET EXPERIMENT, SESSION 8	65
FIGURE 24	SERVER INTER-ARRIVAL TIME IN INTERNET EXPERIMENT, SESSION 9	66
FIGURE 25	CLIENT PACKET SIZE IN INTERNET EXPERIMENT	67
FIGURE 26	SERVER PACKET SIZE IN INTERNET EXPERIMENT	68
FIGURE 27	SERVER INTER-ARRIVAL TIME WITH BANDWIDTH 2.5 MB	69
FIGURE 28	SERVER INTER-ARRIVAL TIME WITH BANDWIDTH 2 MB	70
FIGURE 29	SERVER INTER-ARRIVAL TIME WITH BANDWIDTH 1.5 MB	70
FIGURE 30	SERVER INTER-ARRIVAL TIME WITH BANDWIDTH 1 MB	71
FIGURE 31	SERVER INTER-ARRIVAL TIME WITH BANDWIDTH 0.5 MB	71
FIGURE 32	ANALYTICAL SERVER PACKET SIZE DISTRIBUTION	73

LIST OF TABLES

TABLE 1	COMPARISON OF GAME QUAKE TRAFFIC METRICS IN PUBLISHED RESULTS	6
TABLE 2	COMPARISON OF GAME COUNTER-STRIKE TRAFFIC METRICS IN PUBLISHED RESULTS	6
TABLE 3	NS2 SIMULATION OUTPUT FILE FORMAT	38
TABLE 4	PERCENTAGE AT 1-MS INTERVAL AT PEAKS OF SERVER INTER-SEND TIME	50
TABLE 5	PERCENTAGE OF CLIENT PACKET SIZE OUT OF TOTAL PACKETS	51
TABLE 6	LOCATION AND SCALE PARAMETERS OF EXTREME DISTRIBUTION	54
TABLE 7	SERVER PACKET SIZE TEST RESULTS	55
TABLE 8	REPEATED SERVER PACKET SIZE TEST RESULTS	56
TABLE 9	AVERAGE SERVER PACKET SIZE TEST RESULTS	57
TABLE 10	BASIC STATISTICS OF TRAFFIC IN TEST-BED EXPERIMENT	57
TABLE 11	PARAMETERS OF OTHER TRAFFIC IN SIMULATION	74
TABLE 12	MANN-WHITNEY TEST OF SERVER INTER-ARRIVAL TIME WITH DIFFERENT TRAFFIC	75
TABLE 13	SERVER-TO-CLIENT DELAY WITH DIFFERENT TRAFFIC	75
TABLE 14	MANN-WHITNEY TEST OF SERVER PACKET SIZE WITH SOLE PARAMETER	78
TABLE 15	MANN-WHITNEY TEST OF PSEUDO CLIENT PACKET SIZE	81
TABLE 16	NUMBER OF NON-COMPLIANT OF MANN-WHITNEY TEST OF CLIENT INTER-SEND TIME	82
TABLE 17	MANN-WHITNEY TEST OF PSEUDO CLIENT INTER-SEND TIME	82
TABLE 18	MANN-WHITNEY TEST OF PSEUDO CLIENT INTER-ARRIVAL TIME	82
TABLE 19	MANN-WHITNEY TEST OF PSEUDO SERVER PACKET SIZE BY EXTREME DISTRIBUTION	83
TABLE 20	MANN-WHITNEY TEST OF PSEUDO SLOT-BASED SERVER PACKET SIZE	84
TABLE 21	MANN-WHITNEY TEST OF PSEUDO SERVER INTER-SEND TIME	85
TABLE 22	MANN-WHITNEY TEST OF PSEUDO SERVER INTER-ARRIVAL TIME	85
TABLE 23	CALCULATION OF BINOMIAL DISTRIBUTION	103

GLOSSARY

CDF:	Cumulative Distribution Function.
ECDF:	Empirical Cumulative Distribution Function.
FPS:	First Person Shooter Game. The general objective of FPS games is to explore a common virtual world and kill as many of the other players as possible.
FTP:	File Transfer Protocol is a communications protocol governing the transfer of files from one computer to another over a network.
HTTP:	Hyper Text Transfer Protocol. A protocol used to request and transmit files, especially web pages and webpage components, over the Internet or other computer network.
IETF:	Internet Engineering Task Force (IETF) is a non-membership, open, voluntary standards organization dedicated to identifying problems and opportunities in IP data networks and proposing technical solutions to the Internet community.
IP:	Internet Protocol (IP) is a connectionless, best-effort packet switching protocol. It provides packet routing, fragmentation and re-assembly through the data link layer.
K-S Test:	Kolmogorov-Smirnov goodness-of-fit test.
LAN:	Local Area Network.
MMORPG:	Massively Multiplayer Online Role Playing Games, MMORPG provides some mechanism for character advancement, large areas of landmass to travel across, and other players to interact with.
NNTP:	Network News Transfer Protocol (NNTP) is an Internet application protocol used primarily for reading and posting Usenet articles, as well as transferring news among news servers.
PDF:	Probability Density Function.
RTS:	Real Time Strategy multiplayer online games. RTS games are generally characterized by resource collection, unit construction, and battles that consist of large numbers of animated soldiers standing a few feet apart going through the same animated attack motion over and over.
Self-Similarity:	Self-similarity traffic seems to look the same in the large (minute, hour) as in the small (second, millisecond). At every time scale ranging from milliseconds to minutes and hours, bursts consist of bursty sub-periods separated by even less bursty sub-periods.
SMTP:	Simple Mail Transfer Protocol. The standard e-mail protocol on the Internet and part of the TCP/IP protocol suite.
Sniffer:	A program that records all of the traffic that the network card in a computer sees.

- TCP:** Transmission Control Protocol (TCP) is the connection-oriented protocol built on top of Internet Protocol (IP) and is nearly always seen in the combination TCP/IP (TCP over IP). It adds reliable communication and flow-control and provides full-duplex, process-to-process connections.
- TELNET:** TELetype NETwork (TELNET) is the main Internet protocol for creating a connection with a remote machine.
- UDP:** User Datagram Protocol (UDP) is a connectionless protocol which, like TCP, is layered on top of IP. UDP neither guarantees delivery nor does it require a connection. As a result it is lightweight and efficient, but all error processing and retransmission must be taken care of by the application program.
- VoIP:** Voice over Internet Protocol. Any technology providing voice telephony services over IP. The major advantage of VoIP is lower cost, by avoiding dedicated voice circuits.

CHAPTER 1

LITERATURE SURVEY

1.1. INTRODUCTION

The Internet is expanding rapidly in size, users, applications and traffic. There are more and more requests on network traffic. Network traffic analysis explores packet traffic behaviour and shows possible improvements.

Internet delay is caused by transmission medium and also by routing and queuing. Although delay is often tolerable in many applications such as file transfer or Web, it does affect performances for delay-sensitive applications such as IP phone and video streaming. For First Person Shooter (FPS) games, delay is even more important because it can determine who wins or loses a game.

1.1.1. COMPLEXITY OF INTERNET TRAFFIC

Internet traffic research has shown that Internet traffic can be very ‘bursty’ (packets arrive in clumps). Internet traffic has self-similar scaling behaviour, and self-similar traffic seems to look the same in the large (minute, hour) as in the small (second, millisecond). At every time scale, bursts consist of ‘bursty’ sub-periods separated by even less ‘bursty’ sub-periods (Sarvotham 2001). Bursty traffic results in high bandwidth demands (Sharafeddine 2003). The self-similar traffic is difficult to predict, prone to produce congestion, decreases the effectiveness of buffering in Internet routers (Erramilli 1996), and increases queuing delay and packet loss (Park 1996). Though, researchers have not found direct relation how self-similar traffic affects queuing or end-user performance (Borella 2000).

Because of the importance and complexity of network traffic, a global organization, the Internet Engineering Task Force (IETF), was founded to produce protocol

standards and technical documents. IETF aims to improve Internet design, usage, and management. Internet Engineering usually relates to network traffic.

1.1.2. DELAY-SENSITIVE APPLICATIONS

Although most traditional Internet traffic such as HTTP, FTP, TELNET, SMTP and NNTP can tolerate transfer delay, some other Internet applications are delay-sensitive. Examples: IP call (VoIP), video streaming, and multiplayer online games.

VoIP and video streaming have already been well developed. As to VoIP, latency for voice communication should not exceed 200 milliseconds to ensure good voice quality (Varshney 2002). Different techniques are used to improve the performance of VoIP and video streaming. For example, VoIP and video streaming use good compress coding, and buffering (as to VoIP, de-jitter buffering add delay to voice so buffers should be kept small). VoIP has silence (inside conversation) modelling (silence compression). Video streaming tries to get better Quality of Service (QoS) by applying different network service policies to different types of packets inside video streaming.

Network traffic of both VoIP and video streaming are impacted by packet delay, packet loss, packet jitter (one-way delay variation), packet reordering. Simulation has been used to aid the investigation of VoIP and video streaming traffic behaviour.

Similar to VoIP and video streaming, the performance of multiplayer game is also impacted by packet delay, packet loss, packet jitter, etc. Players of some games have to make decisions in a fraction of a second in fierce battlefield situations. For example, the difference between 50 and 150 ms of delay can determine who wins or loses a game, but such difference is sometimes not noticed by users of IP phone call. Game suppliers and players long for the improvement of network traffic. However, the published research of multiplayer game is not so robust as that of VoIP and video streaming. In order to offer better QoS, it is important to understand the traffic characteristics of games.

Multiplayer games make up around half of the top twenty-five types of non-traditional traffic for some Internet links (McCreary 2000) and are predicted to make up over 25% of Local Area Network (LAN) traffic by the year 2010 (Gargolinski 2005).

1.1.3. GAME TRAFFIC AND SERVER HOSTING

When players play a game on the Internet, their computers automatically send data back and forth to create the illusion of a seamless virtual world in which they all coexist at once.

Game player on the Internet does not only need a game server, but also a game server with good features. A server with better internal hardware (for example, a faster processor or more memory) may improve game performance, but the more important factors are in network (for example, short network traffic delay). Game traffic should be delivered faster and more reliably, with limited delay for all players in a game. Game server hosting is therefore very important.

Different traffic competes for network resources on the Internet. Broad bandwidth helps traffic transfer, but traffic on the Internet is dynamic and game traffic is sometimes impacted by other traffic even if bandwidth is broad. Game traffic characteristics are important to game server hosting because they determines game traffic behaviour in network transfer. Game server hosting welcomes research in game traffic.

1.1.4. FIRST PERSON SHOOTER GAMES

Multiplayer online games are divided into three categories: First Person Shooter (FPS) games, Massively Multiplayer Online Role Playing Games (MMORPG), and Real Time Strategy (RTS) games.

- FPS is played in the first person perspective, characterized by exploring a common virtual world and killing as many of the other players as possible. Notable FPS games include Quake, Unreal, Half-Life, Counter-Strike, System Shock, and Halo.
- MMORPG is capable of supporting hundreds or thousands of players simultaneously, and is played on the Internet. Typically, this type of game is played in a giant persistent world. It enables players to compete with and against each other on a grand scale, and sometimes to interact meaningfully

with people around the world. Notable MMORPG games include Everquest and Ultima Online.

- RTS games are generally characterized by resource collection, unit construction, and battles that consist of large numbers of animated soldiers standing a few feet apart going through the same animated attack motion over and over. Notable RTS games include Starcraft, Warcraft III, and Age of Empires III.

FPS game is extremely delay-sensitive and usually more delay-sensitive than other multiplayer online games. It is important to understand FPS game traffic and its interactions with other network traffic.

1.2. DISCREPANCIES IN PUBLISHED RESULTS

It is important to understand the traffic characteristics of games in order to offer good network service. The discovery of characteristics of game traffic can help the design of communication system. For example, Kim et al. (2005) suggested that Transmission Control Protocol's (TCP) delayed ACK mechanism to be modified for next generation MMORPG games, because they found game packet round-trip-time and inter-arrival time per session were affected by delayed ACK.

An analytical model of game traffic characteristics can be obtained by statistical analysis on empirical game traffic data. A network traffic model can be obtained at different levels (Alberto et al. 2005):

- Session/application-level: arrivals of new sessions and their duration.
- Connection/transport-level: arrivals of new traffic flows and their size.
- Action-level: on-off patterns of the packet generation process.
- Packet-level: lowest level of the packet generation process.
- Byte-level: amount of bytes per second.

Different levels of analysis provide different insights into traffic behaviour. Packet-level traffic behaviour is fundamental, for example, network routers and switches operate on packet-level, delay and loss happen at packet-level, and traffic simulators and generators work in packet-level.

Network traffic analysis and modelling usually have two aspects. The first mainly concerns marginal distribution, it estimates the empirical probability or distributional functions and does curve fitting into well-known distributions. The second mainly concerns dependence and correlation, for example, mutual correlation between packet size and inter-arrival time, autocorrelation, and long-range dependence.

Most published research on game traffic was about the marginal distribution. Research has explored the packet-level marginal characteristics of game traffic, using two main metrics: packet size and inter-arrival time. That is, the behaviour of packet size and inter-arrival time were among the main conclusions in previous game traffic research. Readers would expect that different research on the same FPS game should have the same results, however, the results were not the same.

1.2.1. DISCREPANCIES IN PUBLISHED RESULTS FOR THE GAME QUAKE

Quake by ID Software is a very popular game. The aim of Quake is to move throughout the map *fragging* enemy players and scoring points based on the objective of the game mode. When a player's health points reach zero, the avatar of that player is *fraggd*. Soon after the player can re-spawn and continue playing with their health points full again but without any weapons or power-ups previously gathered. The game ends when a player or team reaches a specified score, or when the time limit has been reached.

The milestones in Quake series are: Quake I (May, 1996), QuakeWorld (Dec, 1996) (early QuakeWorld was an update to Quake I, and QuakeWorld engine supports Quake II games), Quake II (Nov, 1997), Quake III (Dec, 1999), Quake IV (Oct, 2005), Quake Mobile (Oct, 2005), and Quake Wars (May 2007).

The network traffic of the game Quake was investigated in previous research. Borella (2000) probably did the first important work in analysing packet traffic of a multiplayer game: his paper was published in 2000 and was one important reference in most other publications of this subject. He chose the games Quake I and Quake II, and explored the statistical distributions of game packet size and inter-arrival time. Joyce (2000) chose the game QuakeWorld and got the statistical characteristics of

packet size and inter-arrival time. Pavlicic and Armitage (2003), and Lang et al. (2004) chose the game Quake III and got characteristics of packet size and inter-arrival time. There were discrepancies among published results about the game Quake and the discrepancies are listed in the next table:

Metric	Borella (2000)	Joyce (2000)	Pavlicic (2003)	Lang (2004)
Server Packet Size (in Bytes)	Range: 70-120 Mean: 77	Range: 50-100	Range: 40-114 Mean: 66	Range: 60-160
Client Packet Size (in Bytes)	Mean: 24	Range: 50-70	Range: 27-52 Mean: 38	Range: 50-70
Server Inter-arrival Time (in ms)	Peak: 37, 50 and 60 Mean: 27	Peak: 14 and 26	Peak: 66 Mean: 155	Peak: 50 Mean: 50
Client Inter-arrival Time (in ms)	Peak: 14, 21 and 25 Mean: 16-41	Peak: 14 and 27	Mean: 73 Range: 60-86	Peak: 11 Range: 10-40

Table 1 Comparison of game Quake traffic metrics in published results

It has been found that these discrepancies exist not only in published results for the game Quake, but also for the game Counter-Strike (Cunter-Strike 2007).

1.2.2. DISCREPANCIES IN PUBLISHED RESULTS FOR THE GAME COUNTER-STRIKE

Counter-Strike is a team-based, tactical FPS game. Counter-Strike pits a team of counter-terrorists against a team of terrorists in rounds of competition won by completing an objective or eliminating the opposing force. Counter-Strike is widely acknowledged as the most successful and popular of the tactical shooter genre. The game Counter-Strike has also attracted researchers. There were published research results about the traffic of the game Counter-Strike. Similar to the game Quake, the behaviour of packet size and inter-arrival time were explored. Again, main discrepancies among the published results are highlighted in the next table.

Metric	Färber (2002)	Färber (2003)	Feng (2002)	LaPoint (2003)
Server Packet Size (in Bytes)	Range: <1500 Mean: 50.4+6.15n *		Range: 0-300	Mean: >1500**
Client Packet Size: Mean (in Bytes)	52	42	40	165

Table 2 Comparison of game Counter-Strike traffic metrics in published results

* *n* stands for the number of players.

** The mean packet size was over 1500 bytes if there were more than eight players.

1.2.3. DISCUSSION ON CAUSES FOR DISCREPANCIES

Researchers have identified various factors contributing to the change of traffic behaviour.

- Borella (2000) found that game Quake's client packet inter-arrival distributions were highly correlated with the CPU speed of the hosts. The rendering speed had a large impact on transmission rate. The slow hosts had significantly higher and more variable inter-arrival times. Further, traffic to different client might be different because "different amount of information was transmitted to each client".
- Färber (2003) found that the game Counter-Strike's mean server packet size increased as the number of active players increased.
- Lang et al (2004) found the server packet size was strongly dependent on the number of players (each additional player resulted in an overall increase in packet sizes) and to a lesser extent on the particular map. Client inter-send was dependent on the map played and on the client computer's graphic card.
- Bussiere (2006) explored the traffic of the game Enemy Territory (the game uses the game Quake engine). The server packet size varied according to different maps and its distribution became wider with more players. The client packet size differed slightly among players. The client inter-arrival time depended on maps and the client hardware (graphics card, CPU) although the server inter-arrival times were fairly constant.

All the above observations were concerned only with network traffic generating instead of transferring. Traffic behaviour would change if some of the above factors were changed, which may offer certain explanation to these discrepancies. However, network traffic behaviour may change during network transfer, different network environments or observation locations may result in different empirical results. For example, the behaviour of inter-packet time in the sending location is likely different from that in the arrival location on the Internet.

If the game traffic behaviour is related to the observation location, other network traffic or different network, then any analytical model of traffic behaviour should differentiate between the "un-interfered" and "interfered" game traffic. Furthermore,

some location-related metric (for example, inter-packet time) in the sending and arrival locations should be analyzed separately. In other words, two metrics (inter-send time and inter-arrival time) should be introduced instead of just inter-packet time.

However, no research has explored the difference of game traffic metric behaviour in different network locations in an experiment. This lacking of empirical results maybe because of researching focus: previous research was mainly focused on the analytical model of aggregated traffic to (and from) all clients in the game. The game clients were usually in different locations on the Internet, therefore the traffic collection location had to be the game server computer.

There is another possible explanation that the above discrepancies did not attract researchers. The game industry develops quickly and there are always new and more advanced games, and researchers are only interested in the newest games. Some different research on the same game was published roughly in the same time, the researchers therefore did not have comparison from other published empirical results.

1.2.4. INTER-SEND VERSUS INTER-ARRIVAL TIME

Inter-packet time (or inter-arrival time) is one of the basic packet-level metrics of the network traffic. Apart from the games Quake and Counter-Strike, inter-arrival time of some other games was also explored:

- Joyce (2000) found the inter-arrival time characteristics of the game Unreal Tournament's traffic: client's peaked at 25 ms, server's peaked at 60 ms.
- Bangun (2000) found that client packet inter-arrival time was a split distribution composing of two sections, both sections were exponential distributions for the game Starsiege Tribes.
- Heyaime (2002) found that packet inter-arrival time was either Gamma or uniform distributions for four different games.
- Lang and Armitage (2003) found that packet inter-arrival time had very simple traffic pattern and was highly periodic for the game Halo.

Again, similar to the research on Quake and Counter-Strike, the above research was focused on exploring packet inter-arrival time, the difference between the inter-send and inter-arrival time remained unknown.

There was confusion of terminology in previous published research. The term “inter-arrival time” referred to the term “inter-send time” by some researchers. For example, Feng et al. (2002) collected and analysed traffic traces from a Counter-Strike’s server, the inter-packet time of traffic going out from the server was in fact inter-send time, but it was simply regarded as inter-arrival time. Similar misuse was found in Borella (2000). The misuse was not mistake because these researchers were only interested in offering an analytical model of inter-packet time. However, the misuse will not be allowed in exploring the change of traffic behaviour with impact from other traffic.

1.2.5. "CLEAN" VERSUS "INTERFERED" TRAFFIC

The other traffic in this dissertation is all traffic except the game traffic along the same network path, and competes for network resources with the game traffic. In an “un-interfered” network without other traffic such as an isolated LAN, the game traffic is therefore “clean”. The “clean” traffic characteristics represent the traffic generated by a game. On the other hand, if similar experiment is repeated on the Internet, there is other traffic simultaneously and the game traffic is “interfered”. The “interfered” traffic characteristics represent the game traffic with impact from other traffic.

There were discrepancies among the published empirical results in previous sections that were based on “interfered” network. One of the causes to the discrepancies might therefore be different “interfered” networks, or different other traffic. In order to obtain a full understanding of the discrepancies on game traffic, two similar experiments are needed: one experiment in “clean” network to explore non-network factors to the discrepancies and another experiment in “interfered” network to explore network-related factors to the discrepancies. The comparison of the empirical results from both experiments provides insight into game traffic behaviour with effect from other traffic. This part of the work was one of the research objectives.

1.2.6. OBJECTIVE ONE

As discussed above, causes to the discrepancies of the published results had not been properly explored. The starting point for this research was to explore traffic characteristics (using the metrics of packet size and inter-packet time) that are generated by a FPS game. The first objective for this research was to explore:

- The “clean” game traffic and the relation of traffic caused by different server settings.
- The “interfered” game traffic and the relation of traffic caused by different “interfered” network transfer paths.
- The relation between the “clean” and “interfered” game traffic.
- The relation of game traffic in different empirical observation locations, and the relation between packet inter-arrival and inter-send times.

1.3. IMPACT ON GAME TRAFFIC FROM OTHER TRAFFIC

The game traffic behaviour is likely affected by other traffic along the transfer path as the network traffic of one application is often affected by traffic from other applications because they compete for network transfer resources. Major previous research in this area is discussed in this section.

1.3.1. INTERACTIONS BETWEEN GAME TRAFFIC AND OTHER TRAFFIC

1.3.1.1. TCP AND UDP PROTOCOLS

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are two of the core network protocols.

TCP is a connection-oriented protocol. Applications using TCP can create connections to one another, over which they can exchange streams of data. The protocol

guarantees reliable and in-order delivery of data from sender to receiver. TCP also distinguishes data for multiple connections by concurrent applications running on the same host. TCP is the intermediate layer between the Internet Protocol (IP) below it, and an application above it, in network protocol model. TCP is often referred to as TCP/IP (TCP over IP). Applications often need reliable pipe-like connections to each other, whereas the Internet Protocol does not provide such streams, but only best effort delivery and unreliable packets. TCP does the task of the transport layer in network protocol model.

Applications using UDP can send short messages known as datagrams to one another. Compared to TCP, UDP does not provide the reliability and ordering, datagrams may arrive out of order, duplicated, or go missing without notice. UDP is faster and more efficient for many lightweight or time-sensitive transfers without the overhead of checking whether every packet correctly arrived. UDP stateless nature is useful for servers that answer small queries from huge numbers of clients. UDP is required for broadcast and multicast. Voice over IP (VoIP) and online games use UDP.

1.3.1.2. INTERACTIONS BETWEEN GAME TRAFFIC AND OTHER TRAFFIC

Previous research found that network traffic of different applications along the same network path affected one another. First, game traffic affected traffic of other application, for example, game traffic affected TCP traffic's transfer:

- Joyce (2000) found that game traffic did impact the throughput of TCP on a heavily loaded link. Any type of UDP traffic limited the amount of TCP traffic, and games affected TCP in much the same way as VoIP. The underlying UDP protocol behaved in much the same manner (impact the throughput of TCP) regardless of the application that was running it.

Second, other traffic affected the game traffic behaviour conversely, examples are:

- Bangun (2000) found various general network utilization levels affected mean game delay and buffer occupancy. Game delay was affected by other traffic in busy network links.

- Jehaes (2003) found that the other traffic by gaming user (or any another PC on the user's home network) in broadband access networks might saturate the game user's link and impact game delay. However, because the traffic by other users of the same access network was only mixed with the game traffic on high-speed links, it did not increase the game delay unwieldy, if the load on this access network was not too high.

Based on the research of Bangun (2000) and Jehaes (2003), other traffic affected the game traffic delay especially in busy network links. Game traffic is likely to meet busy network links because games are played nearly everywhere on the Internet. However, different network links maybe dominated by different types of network traffic. It is important to investigate further how different traffic affect game traffic behaviour.

1.3.1.3. GAME PACKET DELAY AND INTER-ARRIVAL TIME

Packet delay is important to online multiplayer game performance. Previous research has shown that traffic delay is tightly related to game performance:

- Armitage (2003) found that Quake III users actively preferred servers less than 150 to 180 ms delay from player's location. The delay and game server's location were correlated, farther away players usually have disadvantages in gaming. Distance and server location affected the delay of gaming.
- Nichols (2004) explored the sport game Madden NFL and found that longer delay would cause slower player actions. Although Madden NFL itself can compensate for latency, there was a significant impact on user performance for delay over 500 ms.
- Färber (2002 and 2003) listed various traffic delay ranges versus game player behaviour. Delay did impact game performance.
- Pantel et al. (2003) found that delay of 50 ms was ideal whereas delay over 100 ms should be avoided for car-racing game.

Based on the above results, the game network traffic delay affects game performance directly. Does any other metric of game traffic affect game performance directly?

In gaming, a game host computer receives network packet, processes it, control the game process, and send next packet out. If inter-arrival time of some consequent packets is not regular and very short, it consumes much CPU time (of slow computers) in short interval and likely degrades the game performance, and the last packet likely invalidates un-processed previous-arrived packets. In case that inter-send time is fixed, the extremely short inter-arrival time of some consequent packets likely results in extremely long inter-arrival time of some other consequent packets. The game performance is also degraded by extremely long inter-arrival time because of delay of updating packets. Inter-arrival time directly affects the game performance. It is good to game performance that inter-arrival time is similar to inter-send time, or packets arrive in relatively fixed interval.

Game packet delay and inter-arrival time are two important metrics, and their behaviour is tightly related to other traffic that competes for network resources. However, among researchers referred to in this dissertation, only Bangun (2000), Jehaes (2003) and Carrig (2005) explored the impact on game traffic delay from other traffic, but their work did not concern with different types of other traffic. As long as other traffic does affect the game traffic delay, it is possible that different types of traffic affect game traffic delay differently. The relation between game packet inter-arrival time and other traffic has not been found in published research.

Sometimes different network connections or links maybe dominated by different types of traffic. The route or path selection of game traffic likely need to consider the traffic type on a network link, if there is a noticeable difference of impacts from other traffic on game traffic metrics such as delay and inter-arrival time. Therefore it was one of the research objectives. To gain a better understanding of the game traffic behaviour with other traffic that competes for network resources, the impacts on these two metrics from different types of traffic was explored.

1.3.2. ROLE OF OTHER TRAFFIC IN THE CHANGE OF GAME TRAFFIC

Because of the buffering mechanism of modern network, other traffic can change game traffic inter-packet time because the former and the latter compete for network transfer bandwidth. Game packets have to be buffered and postponed for transfer if

network link is busy, there is likely big difference between game packet inter-send and inter-arrival behaviour.

As previously discussed, there were discrepancies of game traffic behaviour in previous published results, and game traffic behaviour was dependent on several factors such as computer hardware (Borella 2000, Lang et al. 2004, Bussiere 2006), number of players (Lang et al. 2004, Färber 2003, Bussiere 2006), game maps (Lang et al. 2004, Bussiere 2006), observation location and different network other traffic. Among these factors, some are major factors and some are minor ones. Is other traffic that competes for network resources the major factor of the discrepancies? If yes, the challenge is therefore to design an experiment to show the progression from the “clean” to “interfered” game traffic behaviour. Again no research to date has provided answer to this question.

1.3.3. OBJECTIVE TWO

The second objective of this research was to explore the role of other traffic that competes for network resources. It was to build an experiment in order to show the role of other traffic in the progression from the “clean” to the “interfered” game traffic behaviour. It was also to explore the impacts on game traffic metrics from different types of traffic.

1.4. AN ANALYTICAL TRAFFIC MODEL

Game traffic analytical distribution can help realistic network analysis, and traffic models can be used in analytical frameworks, performance evaluation or simulation/emulation scenarios.

A number of traffic analytical models for a few FPS games were obtained in the previous research, and these models reflected the distributions of the game traffic behaviour. However, most researchers referred to in this dissertation did not try to make use of their analytical models in experiment except Lang et al. (2004). Lang generated pseudo game traffic in a simulator according to the analytical model of packet size and inter-packet time. However, Lang did not provide any evaluation of

the usability of the analytical traffic model. For example, Lang did not speculate if the analytical game traffic could replace the realistic game traffic.

If pseudo traffic generated by an analytical distribution model can be used as game traffic, realistic human players can be replaced by the pseudo game traffic in experiments, some game traffic experiments become simpler. In order to achieve this, two consecutive steps were required. The first is rather theoretical and is concerned with constructing a suitable model with reasonable analytical distributions for the traffic metrics; the second is more empirical and was to evaluate the effectiveness of the model.

1.4.1. OBJECTIVE THREE

The third objective of this research was to evaluate the usability of an analytical traffic model, and the possibility of replacing realistic game traffic with the pseudo traffic generated by the analytical model.

1.5. GAME SELECTION

This research was to explore the traffic generated by a FPS game, which game should be selected?

1.5.1. DOUBTS ON CLASSIC WORK

Borella's (2000) work is probably the first in exploring the analytical distributions of game traffic metrics, and is important because it contributed to most other research in game traffic analysis. The work is therefore classic and is still one of the deepest research in analyzing traffic of the game Quake. However, doubts can still be raised on that classic work.

First, server's traffic to all clients were analysed aggregately instead of separately in Borella's work, although he intended to offer "micro" statistical characteristics about

game traffic. The aggregated game traffic was surely important, but if server's traffic to different clients was analysed separately, it could at least tell if server's traffic to different clients had the same analytical model.

Second, Borella only presented results of the most empirical data and discarded a portion empirical data without explanation. For example, there were five clients in the experiment of Quake II, but he only presented the results of four clients in his publication and ignored one client. Besides, some empirical results did not comply to his conclusion, for example, one client had different inter-arrival time distribution from three other clients. He did not provide any comments on this discrepancy from the results, whether it was caused by game server setting, computer hardware, player skill, or other traffic in the network. The similar discrepancy was also in his empirical results of Quake I.

Third, there was a problem in treating the outliers and tail behaviour (the plot outside the bell-shape was called the tail) of empirical data. His solution was ignoring (deleting) them because the magnitude of this heaviness of tail/outlier was minor. Although traffic empirical data are usually large and have outliers therefore it is difficult to find the analytical statistical distribution, deleting a small portion of the original empirical data is not good in data analysis.

Besides, Borella's work was on Quake I and II, they were all old games. It was unknown if a newer Quake game had the same analytical distribution. Based on these doubts, the game Quake was worth further exploring.

1.5.2. QUAKE III ENGINE AND QUAKE III ARENA

A game engine is the core software component of a computer or video game or other interactive application with real-time graphics. It provides the underlying technologies, simplifies development, and often enables the game to run on multiple platforms such as game consoles and desktop operating systems.

1.5.2.1. QUAKE III ENGINE

According to Färber (2003), most FPS games were based on either Quake engine or Unreal engine, and the game Quake III is very fast. Why is Quake III engine fast? It is because Quake III engine supports multi-processing (multi-process).

Both process and thread are kernel scheduling units in computer operating system. A process is the actual execution of a collection of instructions of a program. Processes are typically independent, carry state information, have separate address spaces, and interact only through inter-process communication mechanisms. Processes own resources (for example, memory, file handles, sockets, device handles, and windows) allocated by operating system.

A thread is a thread of execution. Threads are a way for a program to split itself into two or more simultaneously running tasks. Threads do not own resources except for a stack and a copy of registers. At least one thread exists within each process. If multiple threads can exist within a process, they share the state information, same memory and other resources of a single process. Context switching between threads in the same process is faster than context switching between processes.

Multi-processing is the execution of multiple concurrent processes in a system, as opposed to a single process at any one instant. It distributes separate processes to different processors, it need a dual-processor (or multi-processor) computer system.

Multithreading splits threads of a process to either separate processors or a one processor with different logical processors (for example, Intel Pentium IV Hyperthreading processor). Multithreaded program runs faster on computer systems that have multiple CPUs, CPUs with multiple cores (two or more independent CPU cores in a single package), or across a cluster of machines.

Quake III engine is very quick and supports multi-processing. The advantage of Quake III engine is high speed on dual-processor (or multi-processor) computer systems, but the speed is dependent on the specific computer architecture.

1.5.2.2. SELECTION OF QUAKE III ARENA

Based on that the Quake engine is important, the discrepancies of Quake traffic in previous published results and the doubts in Borella's (2000) classic work need to be further explored, Quake was selected in this research.

The Quake III engine is a substantial improvement from its previous Quake engines. Quake III Arena is the most important in the Quake III games. Quake III Arena was specifically designed for multiplayer. This means Quake III Arena allows Internet players to play against one another in real time.

Quake III Arena was selected as the FPS game for this research.

1.6. OBJECTIVES OF THIS RESEARCH

The general objectives in this research were to explore the traffic behaviour of the FPS game Quake III Arena in the next ways:

- To explore the statistical distributions of the “clean” game traffic metrics (a part of Objective One).
- To explore the statistical distributions of the “interfered” game traffic metrics (a part of Objective One).
- To explore other traffic's role in the change from the “clean” to the “interfered” game traffic (a part of Objective Two).
- To explore the impacts on the game traffic from different types of traffic (a part of Objective Two).
- To explore the possibility of replacing realistic game traffic by the analytical distribution model (Objective Three).

CHAPTER 2

METHODS

2.1. OVERALL PLAN OF RESEARCH

The overall plan of this research was to explore the traffic behaviour by a FPS game. It was divided into three experiments.

- A test-bed experiment to investigate “clean” game traffic characteristics: game was played in an isolated local network (test-bed) with no other traffic, game traffic was collected, traffic metrics were parsed and analyzed. The game traffic behaviour was compared among different players, and among different server settings. The statistical distributions of packet size and inter-arrival time were obtained. This is discussed in more details in Section 2.2 and Section 2.3.
- An Internet experiment to investigate “interfered” game traffic characteristics: game was played in realistic Internet, game traffic was collected, traffic metrics were parsed and analyzed. The game traffic behaviour was compared among different servers, and among different observing locations (that is, at source or sink locations). The relation between “clean” and “interfered” game traffic was explored, the effect on the game traffic from other traffic was observed. This is discussed in more details in Section 2.4.
- A simulator experiment to investigate game traffic behaviour with other traffic: game traffic and other traffic competed for network resources in a simulated network, the effect on the game traffic from other traffic was explored. The questions such as game traffic progression from “clean” to “interfered” behaviour, the impacts from different traffic, and the applicability of replacing the realistic game traffic by theoretical model, were answered with empirical data. The causes to the impacts on the game traffic from other traffic were investigated. This is discussed in more details in Section 2.5.

2.2. TEST-BED EXPERIMENT

The objective of the test-bed experiment was to explore the game traffic characteristics without other traffic, that is, “clean” game traffic. An isolated local area network was used for the experiment.

2.2.1. DESIGN OF TEST-BED EXPERIMENT

2.2.1.1. AN ISOLATED TEST-BED

The test-bed for the “clean” game traffic experiment was a LAN of Ethernet. In order to eliminate non-game traffic, the LAN was isolated from other network connection. All computers worked on Microsoft Windows 32 platform. All computers in the LAN except the LAN server were used for the gaming. Among them, one computer served as game server, the remaining computers served as game players’ clients.

Like many other FPS games, Quake III uses client-server architecture, all players’ clients are connected to a single server. There is no direct network communication among clients. The system configuration of game server, game clients, and traffic collecting program (that is, packet sniffer) in the test-bed experiment are shown in Figure 1.

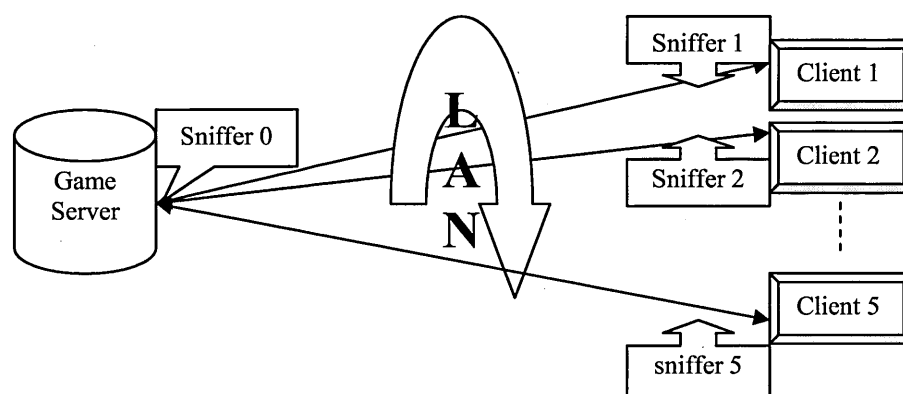


Figure 1 Communication among hosts, clients and sniffers

2.2.1.2. GAME PARTICIPANTS, DURATION AND SESSIONS

In order to ensure typical game traffic, participants were selected to ensure that all game players were skilful in the experiment.

The game traffic is dynamic and random (for example, it is tightly related to players' actions), game traffic can be different in different periods of gaming even with the same players. In order to obtain stable traffic characteristics, game experiment lasted quite a few minutes to ensure the empirical traffic data should be large enough, for example, every client sent (or received) 10000 packets.

The game Quake III offers several configuration selections, for example:

Quake offers "dedicate" and "undedicate" modes, the difference between these two modes is whether game server computer is also served as a game client. Because the traffic collecting program competed for the CPU resource with the game program, in order to decrease the burden on the CPU of game server computer, "dedicate" mode was selected and therefore game server computer did not accept player's input from keyboard or mouse.

Quake III offers a series of maps that consist of different characters in the game. Once a map is selected in game server, all players fight one another in that map. The test-bed experiment was repeated more times (sessions) with the same players but different maps, and the traffic behaviour among different sessions was compared.

2.2.1.3. PACKET SNIFFER

In order to collect network traffic trace for further data analysis, a traffic recording program or packet sniffer is essential. Packet sniffer is a program that records all of the traffic that the network card in a computer sees. Packet sniffers typically modify the kernel of a PC or workstation's network interface to give it packet-capture capability. Such programs save and analyse network traffic.

Ethereal is a free packet sniffer and network protocol analyser for both UNIX and Windows. It examines traffic data from a live network or from a record file on disk. It can interactively browse the recorded data, viewing summary and detail information

for each packet. Ethereal's time precision is one micro second (this time precision is good among packet sniffers and some other sniffer does not offer this precision). In order to use data in sniffed trace-files, some programming is necessary. Ethereal is open source (whereas many other packet sniffers are not) and has a mailing list which offers help to users. The game traffic researcher Lapoint (2003) made a comparison among well-known packet sniffers and recommended to use Ethereal.

Tcpdump is the most widely used network sniffer/analyser for UNIX, and was used by Borella (2000) and Bangun (2000) in their game experiments. Several packet sniffers including Ethereal use Tcpdump as a back-end. WinDump is the porting to the Windows platform of Tcpdump. WinDump is fully compatible with Tcpdump and runs under Windows. The packet sniffer Ethereal uses the support of WinPcap, which is the architecture for packet capture and network analysis of WinDump. Ethereal was selected for the research.

In order to record game traffic, the packet sniffer Ethereal was installed and ran on game server host computer and every game client host computer. Ethereal automatically recorded every packet and its time on a trace-file. The time of every packet was the packet's sending time (if the traffic came out from a game computer) or the packet's arrival time (if the traffic entered into a game computer).

2.2.2. DATA COLLECTION OF TEST-BED EXPERIMENT

The most essential metrics of network traffic are packet size and time (or inter-packet time). The objective of the test-bed (and Internet) experiment was to explore the characteristics of packet size and inter-packet time of the game traffic.

2.2.2.1. PARSING ETHEREAL FILE

The Ethereal disk file contained all information of a game traffic trace. The empirical data of the required metrics should be parsed by programming. For example, the disk file only had time for every packet, but the inter-packet time (instead of time) was the expected metric. And the disk file on server's host computer contained inbound and

outbound packets, these inbound packets consisted of packets from different clients and these outbound packets consisted of packets to different clients, but packets to or from individual client were analysed separately. These problems were solved in the parsing program. The empirical data of packet size and inter-packet time were then converted into the format of histogram.

The game Quake uses UDP protocol for network communication. The metric of packet size in this dissertation is pure UDP data (excluding UDP header).

2.2.2.2. DELETING SOME DATA

This experiment was to explore game traffic characteristics, all other traffic was deleted and only game traffic was parsed from Ethereal disk file.

The game Quake III allows player to change default setting for the gaming (for example, selecting a special weapon, or adjusting brightness in a map, etc). Some skilful players in this experiment had their preferences. They changed default settings while being connected to the game server before they started fighting in the game, others did not. Players started fighting in the game at different time. Once a player is connected to a game server, lots of packets keep being exchanged between server and this player's host computer. The time used in personal setting (for example, one minute) can be relative long in a game experiment which lasts only minutes. Lots of network packets are generated and transferred in this gap. When parsing data from the Ethereal trace-files, network packets before all players started fighting in the game were deleted in order to ensure the typical "gaming" network traffic.

Although different players had different starting-times, the ending time was same because the game ended when the game server exited.

2.3. DATA ANALYSIS OF TEST-BED EXPERIMENT

After the empirical data of a traffic metric were obtained, an analytical distribution was determined from the empirical data's statistical characteristics using curve fitting

technique. Curve fitting is to find a curve (that is, analytical function) which matches the empirical data, and the parameter values of the analytical function that most closely match the empirical data.

2.3.1. GOODNESS-OF-FIT TEST

Goodness-of-fit tests are traditional tools for curve fitting.

Goodness-of-fit tests indicate whether or not it is reasonable to assume that a random sample comes from a specific distribution. Goodness-of-fit tests are a form of hypothesis testing where the null and alternative hypotheses are:

H₀: Sample data come from the stated distribution.

H_A: Sample data do not come from the stated distribution.

However, the failure of goodness-of-fit tests in network traffic analysis was noticed in previous research. According to Paxson (1996) and Bellera (2000), the traditional goodness-of-fit tests often failed in fitting Internet traffic to analytical distribution, because network traffic traces were both large (thousands of samples per traffic trace) and highly variable.

The two widely-used goodness-of-fit tests are Chi-square test and Kolmogorov-Smirnov (K-S) test.

2.3.1.1. CHI-SQUARE TEST

In Chi-square test, the empirical data should first be grouped. The actual number of observations in each group is compared to the expected number of observations and the test statistic is calculated as a function of this difference. The number of groups and how group membership is defined will affect the power of the test. Power will also be affected by the sample-size and shape of the empirical distributions.

Chi-square test was tried with the empirical data in this research. Unfortunately, it did not get a fitting analytical distribution (that is, every null hypothesis was rejected) no matter how the empirical data were grouped. The major cause for the rejection was that heavy upper tail (“upper” indicates right side of the curve according to the

increasing direction of the x coordinate axis, and “tail” means the curve outside the bell shape) yielded very large Chi-square statistic value, according to the details in the Chi-square test procedure. Therefore Chi-square test was not used in the research.

2.3.1.2. KOLMOGOROV-SMIRNOV TEST

Unlike Chi-square test, the K-S test is based on the cumulative distribution function (CDF) instead of the probability density function (PDF). More accurately, K-S test is based on empirical cumulative distribution function (ECDF).

Given N ordered data points: Y_1, Y_2, \dots, Y_n , the ECDF is defined as

$$E_n = \frac{i}{N} \quad (1)$$

where i is the number of points less than Y_i , and every Y_i is ordered from the smallest to the largest value.

The K-S test statistic value is defined as

$$D = \max_{1 \leq i \leq N} \left| F(Y_i) - \frac{i}{N} \right| \quad (2)$$

where $F(Y_i)$ is the analytical cumulative distribution function which must be a continuous distribution, and it must be fully specified.

The statistic D in Formula (2) is calculated and compared to the value V in Formula (3) at every ordered point. D should be no more than V at every ordered point for a non-reject hypothesis.

$$V = \frac{c}{\sqrt{M}} \quad (3)$$

where c is a constant but depends on the confidence level, c is given as 1.22 and 1.36 on confidence levels of 5% and 10%, respectively. M is sample-size.

K-S test was not directly applicable to the empirical data in this research because network traffic consisted thousands of packets. The large sample-size resulted in small value of the statistic V according to Formula (3). The small V value in turn resulted in rejects in every K-S test.

2.3.2. CURVE FITTING PROGRAM

With the development of computer techniques, some statistical programs offer curve fitting function together with parameter determining. Such programs can solve some complicated curve fitting tasks.

Although there are many statistical programs, they have limitation. For example, the widely-used plots fitting programs Matlab and Gnuplot cannot solve the specific curve fitting problem in this research.

2.3.2.1. MATLAB

Matlab is a powerful, easy-to-use package, used interactively on many computer systems to provide data manipulation and statistical analysis. Matlab runs on PC and most workstations, minicomputers and mainframe computers. Matlab provides a user interface that makes statistical analysis more intuitive for all levels of users. Pull-down menus and dialog boxes give users easy prompts in every step.

The Curve fitting Toolbox of Matlab provides a library of linear, nonlinear, and nonparametric fitting models, including:

- Polynomial
- Exponential
- Rational
- Peak (Gaussian)
- Distribution (Weibull)
- Fourier and power series
- Spline (cubic and smoothing)
- Interpolant (linear, nearest-neighbour, cubic-spline, shape-preserving)

Despite its power, Matlab does not support curve fitting of distribution with user-defined formula, for example, it does not support curve fitting of extreme distribution. Therefore Matlab was not selected in data analysis.

2.3.2.2. GNUPLOT

Gnuplot is a free, command-driven interactive function and data plotting and fitting program. The curve fitting in Gnuplot can provide fully defined fitting function for empirical data. Gnuplot provides curve fitting together with qualitative deviations estimating. It does not offer confidence intervals, but reports parameter error estimates instead. These estimates are called “asymptotic standard errors”. The fit is judged by reduced chi-square, the sum of the squared residuals (WSSR) divided by degrees of freedom. However, as the official Gnuplot online documentation (Gnuplot 2004) stresses, the asymptotic standard error inside Gnuplot is generally over-optimistic, should not be used for determining confidence levels, but can be used for qualitative purposes. Therefore Gnuplot was not selected in determining the curve fitting.

2.3.3. AN APPROXIMATE METHOD OF CURVE FITTING

2.3.3.1. SACRIFICE IN PREVIOUS RESEARCH

The network traffic traces are highly variable, it is usually difficult for them to fit any well-know probability distribution. Researchers sometimes managed to get a fitting analytical distribution by removing (deleting) a tiny portion of the empirical data. An example of such sacrificing data was in the usage of the λ_2 discrepancy method.

The λ_2 discrepancy measure was introduced in (Pederson 1990). It facilitates the determination of how close a data set fits an analytical distribution. Applying λ_2 to network data has been discussed in detail (Paxson 1994). The λ_2 discrepancy method determines the discrepancy between the empirical data and the analytic distribution, instead of determining whether or not a data set fits an analytical distribution.

Paxson (1994) used the λ_2 discrepancy method for Internet traffic, Borella (2000) used the λ_2 discrepancy method for the game Quake’s traffic. The problem is, both of them had to remove (delete) some packets’ empirical data within a traffic trace to obtain a

fitting distribution, that is, they were not able to obtain any analytical distribution without sacrificing a tiny portion of the empirical data.

It is likely that Paxson and Borella did not have other choice with their empirical data. The particular nature of the traffic trace is that they are “spiky” data sets, the empirical distribution cannot be smooth. Such data are difficult to fit to any well-known analytical distribution, even if the analytical distribution fits the general shape of the data set very well. The situation gives rise to a point: Sacrifice in strictness is sometimes necessary in obtaining an approximate analytical distribution.

Assume it is necessary for certain kind of sacrifice, there are different ways of sacrifice. For example, sacrificing (deleting) data is one way, sacrificing the strictness of a modelling method is another way. Sacrificing the strictness of a method means the method is not in 100% confidence mathematically. Unlike the arbitrariness in deleting empirical data, a less-strict method at least offers regular steps to be followed and no empirical data are deleted. The only disadvantage is that the method itself was an approximate method.

A less-strict mathematical method was introduced in data analysis.

2.3.3.2. SELECTION OF KOLMOGOROV-SMIRNOV TEST

The Internet or network traffic usually has heavy upper tails. After visual examination of the empirical plots of the metric of the server packet size, heavy upper tails existed in the empirical data of this research, because the empirical plot was higher than the analytical plot in upper tail portion.

One main limitation of K-S goodness-of-fit test is that it tends to be more sensitive near the centre of the distribution than at the tail. This limitation might be helpful in obtaining the analytical distribution for the empirical data. It might help to get a non-rejected null hypothesis in curve fitting, because the upper tail was the major discrepancy (therefore the major obstacle) in achieving a positive test result.

2.3.3.3. M-TRIAL TEST

The number of packets in any empirical data of the test-bed experiment was large. According to the K-S test's definition, the larger the sample-size is, the more likely the K-S test leads to a "rejected" result. Because of this, it is almost impossible to achieve any "non-rejected" result if K-S test is directly used on a large population.

In order to solve this problem, a small sample-size was used for the K-S test. The method is named the *m-trial test* and its steps are:

- A number n (for example, 100) is set as the sample-size. N samples are randomly selected for from the population.
- A number m (for example, 30) is set as the number of trial. The K-S test with the same sample-size n (for example, 100 in this case) is repeated m times with different randomly-selected samples (the total number of random samples in the *m-trial test* is $n*m$, 3000 in this case).
- Let m be 30 (*m-trial test* became 30-trial test). Every 30-trial test offers a number (between 0 and 30) of rejects. At the 10% significance interval, if the number of rejects is equal or less than 5, the null hypothesis is not rejected (see Appendix A).

More explanations about the *m-trial test* are as follows:

The value of n was set to 100 in the above case. In fact, the selection of the value of n can be different if similar tests were repeated. The sample-size n can be other values such as 36, 49, 64, 81, 100, 121, 144, 169, 196, and 225. These values were selected for simplifying the root calculation in the definition of K-S test (the root calculation is in Formula 3). Different value of sample-size n can offer insight into the *m-trial test* results.

The selection of value of m as 30 was based on obtaining stable test results due to network traffic variability after many tries. Smaller values of m cannot compensate the variability of game traffic in this research.

Different confidence levels (for example, 5%) of the above test were tried with same empirical data. Much higher proportion of null hypothesis was rejected in the confidence level 5% (and certainly also in the confidence level 1%). That is why the

selection of the confidence level was 10%. The confidence level of the results of this statistical m-trial test was therefore 10%.

2.3.3.4. PROCEDURE OF DATA ANALYSIS

According to the previous sections, the overall procedure of curve fitting for game traffic metric was:

- Converting the empirical data of a metric into histogram's format.
- Drawing the PDF or CDF plots of empirical and potential analytical distributions; visually examining the plots and determining the analytical distribution for the empirical plot. Stopping if the statistical distribution can be determined by visual examining plots, otherwise going to the next step.
- Using the curve fitting function of Gnuplot to determine the best analytical distribution and obtain the function parameters of the analytical distribution.
- Determining the fitness between empirical and analytical distribution by the m-trial test in the last section.

2.4. INTERNET EXPERIMENT

The objective of the Internet experiment was to explore the game traffic behaviour with other traffic, that is, "interfered" game traffic.

2.4.1. DESIGN OF INTERNET EXPERIMENT

The design of the Internet experiment was similar to that of the test-bed experiment. The game was played by participant and network traffic was recorded by the packet sniffer Ethereal. In the Internet experiment, the same experiment was repeated many times (sessions), with game server selection of different server-client delay, different game types (Free-for-All, Team-Deathmatch or Catch-the-Flag). The differences among sessions of the experiment were obtained.

The main difference from the test-bed experiment was that players in the same game had different geographical locations and were connected to a game server on the Internet. The empirical game traffic traversed together with other network traffic along the network path(s) between game server and client.

2.4.1.1. SERVER SELECTION

How does a game player find or select a server on the Internet? Like many other FPS games, Quake software contains a game server selection browser. In Internet gaming, when a player starts a game client anywhere on the Internet, the server selection browser in client computer contacts a master server that records all active servers. The client then individually pings each server to get delay and other information (for example, title, battlefield map, number of players). A list of recommended game servers (usually geographical or topological local to the client) is then shown on client computer, the player can join any server in the list and start gaming. With the help of server selection browser, the game server in the Internet experiment was selected in the server selection browser list.

The game server in this experiment was dynamically selected. Some factors in this experiment (for example, other traffic, network path between game server and client) were dynamic or random. Different networks, paths or other traffic may result in different game traffic behaviour. The experiment was therefore repeated many times (sessions) with different game servers. The game traffic with different servers was then compared.

2.4.1.2. CLIENT TRAFFIC

Unlike quite a few participants in the test-bed experiment, there was only one participant in the Internet experiment, the other players in the same game were from somewhere on the Internet. This participant and the other players in the experiment fought one another in the virtual battlefield. In order to compare game traffic in test-bed and on the Internet, this participant used the same computer in the test-bed experiment (the computer was connected to the Internet via the University of Abertay Dundee).

2.4.2. DATA COLLECTION AND ANALYSIS

This participant competed with all other players on the Internet in this experiment. The participant's client computer sent game traffic to a game server on the Internet, and received game traffic from the server. The packet sniffer Ethereal ran in this client computer, game traffic both inwards and outwards was collected. Based on the design of this experiment, the client's traffic was collected at sender's end (therefore the corresponding metric was the inter-send time), the server's traffic is collected at receiver's end (the corresponding metric was the inter-arrival time). Because there was no direct packet communication among clients in the game Quake III, the incoming game traffic was only from the game server, and the outgoing game traffic was only sent towards the game server.

After the game traffic traces were recorded, traffic metrics of packet size and inter-packet time were parsed from the original trace for further analysis.

The data collection and analysis followed the same methods as in the test-bed experiment (see Section 2.3). After the empirical data were parsed, the empirical data of the game traffic metrics of the client and server packet size, the server inter-arrival time, the client inter-send time were obtained. They were converted into histogram, plots were drawn, and distribution was determined. The plots of the "interfered" traffic in the Internet experiment were compared to those of the "clean" traffic in the test-bed experiment.

2.5. SIMULATOR EXPERIMENT

The objective of the simulator experiment was to explore the effects on the game traffic from other traffic.

In order to explore the impact on game traffic from other traffic, two simulator experiments were designed, that is, an experiment to show the progression from "clean" to "interfered" game traffic behaviour, and an experiment to compare the effects on game traffic from different types of traffic.

Another simulator experiment was designed to explore the applicability of the analytical game traffic model. That is, to explore the difference between analytical and realistic game traffic, before and after transferring in simulation network with other traffic.

2.5.1. INTRODUCTION TO SIMULATION

In the realistic Internet experiment, some environment factors (for example, other traffic, path between two game computers, server's location on the Internet) cannot be controlled because they were random or variable. It is difficult to determine the effects on the game traffic from other traffic because of the random or variable factors, even if difference was observed in the "interfered" network from original "clean" game traffic, or among various "interfered" game traffic.

A new experiment was designed to explore the impact on game traffic from other traffic. The experiment was performed with controllable environment factors. The change of the impact was observed by adjusting controllable environment factors in the experiment.

Simulation is a good approach of performing network or Internet experiment with fully controllable factors.

2.5.1.1. NETWORK SIMULATION

The Internet traffic experiments are usually difficult because the Internet is extremely dynamic and complex.

There are direct and indirect approaches to perform network or Internet related experiment. The direct approach is to implement and deploy the experiment in the Internet. For wide-area Internet applications, the direct approach is expensive and difficult. For example, it may not be feasible to obtain the network resources for representing wide-area network effects.

Network emulation is the indirect approach. It eliminates the problems of obtaining remote computational resources and network access. Network simulation runs an actual experiment on a computer or LAN. The experiment on a computer or LAN

offers the behaviour of delay, load, bandwidth, topology, and other network metrics. Network simulator experiment offers an easy approach to network evaluation or traffic behaviour observation.

There are two ways to carry out network simulation. One way is to develop a specific simulator, which requires abundant knowledge of networking, skilful programming and time. The other way is to make use of a general-purpose simulator.

Many published network simulator experiments were based on general-purpose simulators.

2.5.1.2. NETWORK SIMULATOR

Network simulator provides experiment support for network simulation. There have been many such network simulators and they meet lots of simulation requirements. Such network simulators are OpNet, Comnet III, Cnet, NetSim, NetViv, ns2, etc.

Ns2 is one of the most widely used simulators. It is a discrete event network simulator. It is free and has well-supervised mailing lists which offer help to ns-users and ns-developers, respectively. It supports both wired and wireless network simulation, can run under Unix, Linux, and Windows. It is open source and easy to modify. It provides results both in traffic disk-file and visualization interface.

Several published network simulation results were based on the ns2 simulator. For example, two papers (Ke 2005 and Shieh 2001) in IEEE letters and conference proceedings and two papers in ACM conference proceedings (Legout 2000 and Feldmann 1999) were primarily based on ns2 simulation results. It is reasonable to assume that ns2 simulator is correct and reliable.

The simulation in this research was based on the ns2 simulator (ns release 2.29).

2.5.2. PROGRESSION FROM “CLEAN” TO “INTERFERED” TRAFFIC

The packet size and inter-packet time are two essential metrics of network traffic. Packet size will not change during network transfer if there is little packet drop, but inter-packet time can change. Theoretically, when there is no other traffic and

bandwidth was sufficient for the game traffic, the packet inter-arrival time is the same as the inter-send time. Game traffic inter-arrival time becomes different from inter-send time when game packets compete for network resources with other traffic during network transfer. The inter-arrival time is dependent on the competition of the network resources and is likely different with different traffic. The metric of inter-arrival time was selected to show the progression from the “clean” to the “interfered” game traffic with other traffic that competes for network resources. Different inter-arrival time characteristics can likely be observed by adjusting other traffic or network configuration in simulator experiment. The experiment was to build a bridge between the “clean” and “interfered” game traffic.

2.5.2.1. NETWORK CONFIGURATION

A network configuration was designed to perform the simulator experiment. Network nodes represented the traffic sources, mid-way points and sinks of game traffic and other traffic, network links offered paths for network traffic transfer.

A game server and a few client nodes were in the simulator experiment. In the realistic game playing or the game in the previous research, the number of players is usually less than the maximum number of players of a game. The simulator experiment was designed with six clients and one server of the game.

There is no direct traffic transfer among clients in the game Quake III Arena, game traffic is only exchanged between server and individual client. The data collection of the experiment was focused on just one such server-client connection, that is, the network link between the server node (Node 3) and one client node (Node 0). Other traffic was added to that network link, therefore there were two nodes (Node 1 and 2) between the server and that client. The sources and sinks of the other traffic were Node 4 and 5. The link between Node 1 and 2 was named the “shared link” because it was the only link in the network that was shared by game and other traffic. The remaining five client nodes (Node 6 – 10) were connected to the server node directly. The network configuration is shown in Figure 2.

In the Internet experiment, a player client should select a game server from a list of active servers on the Internet. The servers were listed together with the delay between

the client and server. Many non-team type (Free-for-All) servers in the list were usually between 35 and 60 ms away from the client. Game players would like to select server with less delay. Therefore the delay between the server and that client in the simulator experiment was set to a value between 35 and 60 ms (44ms in the configuration).

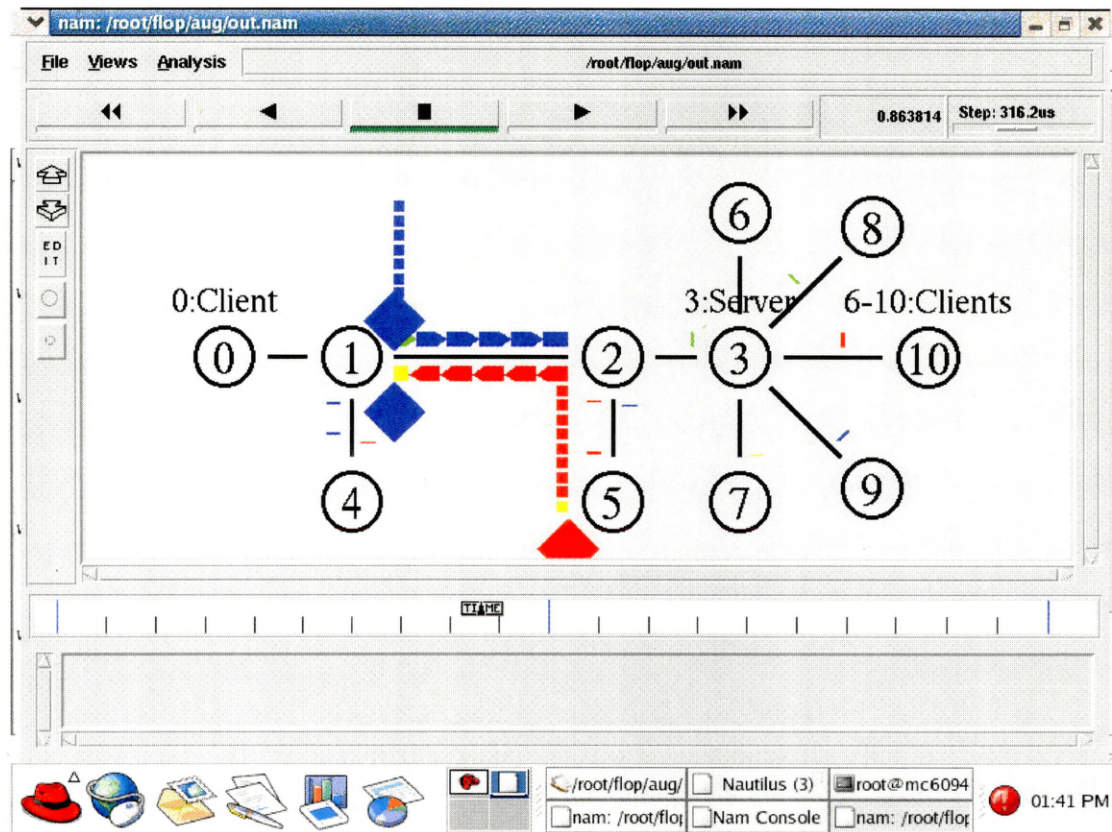


Figure 2 Network configuration of simulation

The icon-shaped markers in Figure 2 represent transferring packets on the link. The vertical queues of markers beside Node 1 (or Node 2) represent buffered packets in the queue, the diamond markers represent dropped packets from the queue. The “shared link” was busy while the screenshot was taken because the link was transferring several packets (several arrowed icons) along both directions.

2.5.2.2. GAME TRAFFIC BY ORIGINAL TRACE

Game traffic was required and therefore generated in the simulator experiment.

In the test-bed experiment, the game traffic was recorded into trace-file on host computer before traffic entered into the LAN test-bed. The traffic trace that came immediately out from a game computer was typical source game traffic, it was used as the source game traffic (by generating/replaying game traffic due to packet size and inter-packet time) in the simulator experiment.

The packet sniffer Ethereal offered several game traffic trace-files in the test-bed experiment. Ns2 simulator can generate network traffic according to a trace-file. However, the ns2 simulator cannot directly use Ethereal trace-file. Ns2 source traffic trace-file requires a specific format. Every record in ns2 source trace-file consists of two fields: inter-send time and packet size. The inter-send time and packet size must be translated into binary format before they are written into any ns2 source trace-file. A C++ program was developed to convert game traffic trace into ns2 specific format trace-file (the correctness of the program was validated by collecting the simulation-generated traffic stream, parsing the metrics of inter-send time and packet size from the stream, and comparing the parsed metric to the original metric).

Once source traffic trace-file was ready, it should be attached to a specific type of traffic source (that is “UDP traffic source”) in ns2. The “UDP traffic source” in turn should be attached to one network node. After these steps, ns2 simulator can automatically generate traffic according to the source trace-file.

2.5.2.3. OTHER TRAFFIC GENERATION

Ns2 simulator can automatically generate TCP traffic and UDP traffic. The TCP traffic by ns2 is FTP (File Transfer Protocol) traffic stream, the UDP traffic by ns2 are CBR (Constant Bite Rate), EXP (exponential distribution), PAR (Pareto distribution, see Appendix B for the definition of Pareto distribution). Ns2 itself offers a segment of scripts of generating WEB traffic.

The ns2 TCP traffic is a stream that adjusts its volume due to network parameters. FTP tends to increase its sending rate until a part of the network is congested.

The ns2 CBR traffic is a stream at constant bit rate. Both EXP and PAR traffic inside ns2 simulator are on/off traffic. For example, the Pareto traffic stream is generated according to a Pareto on/off distribution. Packets are sent at a fixed rate during on

periods, and no packets are sent during off periods. Both on and off periods are taken from a Pareto distribution with constant size packets. The difference between EXP and PAR traffic is that the times of on-periods and times of off-periods are taken from exponential or Pareto distribution, and the parameters of exponential or Pareto distribution can be controlled in user program by selecting different values of parameters.

The ns2 WEB traffic is a multi-session stream. It is partly controlled in user program by adjusting number of sessions, inter-session time, and some other networking factors.

As to the experiment to observe the progression from the “clean” to the “interfered” game traffic, all of the above five different types were tried with different parameters and setting, the progression from the “clean” to the “interfered” game traffic was likely different with different other traffic, only the best progression result was selected and shown in this dissertation.

2.5.2.4. RESULT INTERPRETATION

Ns2 simulator not only provides result visually in its animation tool NAM, but also stores the whole result of simulation in an output file. The output file contains every packet event (an event is a packet arrives or leaves a network node or queue) in simulation. The record format of the output file is shown in the next table:

event	time	from node	to node	pkt type	pkt size	flags	fid	src addr	dest addr	seq num	pkt id
-------	------	--------------	------------	-------------	-------------	-------	-----	-------------	--------------	------------	-----------

Table 3 Ns2 simulation output file format

Exactly speaking, a record in output file represents an event instead of a packet. Each record starts with an event (+, -, d, r) descriptor followed by the simulation time (in seconds) of that event, and from and to node, which identify the link on which the event occurred. The next information is packet type and size (in bytes), and the next is flags (looks like “-----” if no flag is set). The next field is flow id (fid) of IPv6 that a user can set for each flow. Even though fid field may not be used in a simulation, users can use this field for analysis purposes. The fid field is also used when specifying stream colour for the simulation display. The next two fields are source and

destination address in forms of “node.port”. The next field shows the network layer protocol's packet sequence number. Even though UDP packets do not use sequence number, ns2 keeps track of UDP packet sequence number for analysis purposes. The last field shows the unique id of the packet.

2.5.2.5. PROGRAMMING TOOLS

A programming language tool was required for instructing ns2 simulator to perform designed simulation. Tcl (Tool command language) is a very powerful but easy to learn dynamic programming language, suitable for a very wide range of uses, including web and desktop applications, networking, administration, testing and many more. Tcl is open source, cross platform, easily deployed and highly extensible. Tcl was used in writing ns2 simulation program in this research.

A programming language tool was required for interpreting the empirical output (result) from ns2 simulator. Awk is a pattern-matching programming language and allows users to perform relatively sophisticated text-manipulation operations. Awk was selected as programming tool in parsing ns2 output files in the simulator experiment.

2.5.2.6. PROGRESSION EXPERIMENT

This simulator experiment was to show the progression from the “clean” to “interfered” game traffic behaviour.

The inter-send time of the test-bed experiment approximately had deterministic distribution. However, the inter-arrival time characteristic in other published research was different from deterministic distribution. Most inter-arrival time plots had a bell-shape around the location of the peak (or peaks) in inter-send time plot, although inter-arrival time plots usually differed from one another. This experiment was to show the typical progression of empirical plot from peak to the bell-shape.

It was unknown which type of traffic (and the selection of parameters of the traffic), or combination of different traffic, can show a typical progression of the metric of inter-packet time. Therefore this simulation required lots of tries before obtaining a

typical progression. Different effects on game traffic from other traffic were achieved by adjusting traffic parameters and the network bandwidth of the “shared link” in the simulator network configuration.

The progression built a bridge between the “clean” and the “interfered” game traffic, and served as a base to explain the discrepancies in the empirical results of previous research.

2.5.3. EFFECTS FROM DIFFERENT TYPES OF TRAFFIC

The game traffic behaviour can be different due to different volumes of other traffic that competes for network resources, or even different types of traffic. The objective of this simulator experiment was to compare effects from different types of traffic.

Because this experiment was similar to the experiment of “progression from clean to interfered game traffic”, the network configuration, game traffic generation, other traffic generation, and result interpretation were the same as in the progression simulator experiment.

2.5.3.1. METRICS

As discussed in the progression simulator experiment, the metric of inter-packet time is affected by other traffic, therefore the effects on the game packet inter-arrival time from different types of traffic were collected and compared in this experiment.

Delay feature is important to games. If there is sufficient network bandwidth along network path for the game traffic and there is not any other traffic, the end-to-end delay of game traffic (server-to-client delay, or client-to-server delay) is the sum of the delay of every link along the path. However, when there is other traffic along the path and the sum of the game and other traffic surpasses bandwidth, network packets are buffered in queue and the end-to-end delay increases. Delay and other traffic are related. Therefore the effects on the game end-to-end delay from different types of traffic were collected and compared in this experiment.

2.5.3.2. OTHER TRAFFIC

In order to compare the effects on time-related metrics (for example, delay and inter-arrival time) from different types of traffic, traffic with different behaviour of inter-packet time but the same in all other parameters (for example, traffic rate, packet size, etc) were used in simulation. The traffic was FTP, CBR, EXP, PAR and WEB. The same simulator experiment was repeated five times, every time with different types of traffic. Except the type of traffic stream, all other parameters of these five streams were kept the same in the experiment.

2.5.3.3. DATA COLLECTION

Once the bandwidth of the “shared link” and the queue length remained unchanged, game traffic behaviour changed due to different congestion levels (that is, different volumes of other traffic) in the “shared link”. As long as link bandwidth is sufficient for game traffic, different volume of other traffic caused different game traffic behaviour: no game packet dropped when other traffic volume was low; game packet-drop appeared when other traffic volume increased; and lots of game packets dropped when other traffic volume was high (the situations can be viewed in ns2’s graphic animation tool NAM).

On one hand, game packets in the experiment should be queued or even dropped in order to observe the impact from other traffic. On the other hand, dropped game packets distort the results of end-to-end delay and inter-arrival time in the game traffic sink (for example, packet-drop can affect inter-arrival time, because some inter-arrival time was not the time between two consequent packets. Inter-arrival time became larger if packets dropped). To balance queuing and dropping, the data (result) collection in the simulator experiment was performed when game packet started to drop (or immediately before game packet drop started to happen). The network parameters (queue length, bandwidth, etc) and the volume of other traffic stream were usually adjusted together to get the situation of game traffic starting to drop.

2.5.3.4. DATA ANALYSIS AND MANN-WHITNEY TEST

The game traffic behaviour was observed and recorded with different traffic in the experiment. One of the recorded metrics was the game traffic inter-arrival time. The game inter-arrival time with different traffic in the experiment was compared.

A simple way to compare two or more data sets is test for a significant difference in their means. However, for some specific data sets (for example, approximately normal distributed), the mean is not an appropriate summary statistic.

The Mann-Whitney test is one of the most powerful nonparametric tests for comparing two populations. It is a test of both location and shape. Given two independent samples, it tests whether one variable tends to have values higher than the other. It is used to test the null hypothesis that two populations have identical distribution functions against the alternative hypothesis that the two distribution functions differ only with respect to location (median). The Mann-Whitney test does not require the assumption that the two samples are normally distributed.

The Mann-Whitney test is used to compare two unpaired groups. The key result is a p-value that answers this question: If the populations really have the same median, what is the chance that random sampling would result in medians as far apart (or more so) as observed in this experiment?

If the p-value is small (less than 0.05), the hypothesis that the difference is a coincidence can be rejected, and conclude instead that the populations have different medians. If the p-value is large, the data do not give any reason to conclude that the overall medians differ.

The Mann-Whitney test can be only used with small sample-size (single less than 80, or both less than 30), but the empirical data of simulation result were large, therefore samples were randomly selected from empirical data for Mann-Whitney test.

Minitab is a powerful, easy-to-use program package, used interactively on many computer systems to provide data manipulation and statistical analysis. Minitab runs on PC and Macintosh computers, most of the leading workstations, minicomputers and mainframe computers. Minitab for Windows provides a user interface that makes statistical analysis more intuitive for all levels of users. Pull-down menus and dialog

boxes give easy prompts every step of the way. Minitab (release 14) provided the Mann-Whitney test for data analysis in the simulator experiment.

2.5.4. ANALYTICAL TRAFFIC MODEL

The game traffic was collected before it started network transfer in the test-bed experiment, and the analytical model of the collected traffic was obtained. Was it possible to use the analytical model to generate pseudo game traffic in experiment? In order to answer this question, data analysis was performed to examine the variation of the pseudo traffic from the realistic traffic. Metrics of pseudo game traffic by the analytical distribution were analysed not only for traffic generation, but also for the change through network transfer. Therefore not only inter-send time, but also inter-arrival time were analysed. When there is no or only a little packet drop during transfer, the characteristics of packet size will not change. Therefore the packet size was only analysed for traffic generation. There were six metrics in the data analysis and they were the server packet size, the server inter-send time, the server inter-arrival time, the client packet size, the client inter-send time, and the client inter-arrival time.

In order to examine the variation of the pseudo traffic from the realistic traffic after network transfer with impact from other traffic and analyse the metric of inter-arrival time by the analytical distribution, a simulator experiment was performed. The simulation had the same network configuration as the rest simulations in this research, but with different selection of the other traffic. Simulator provided the same network and the same other traffic for both pseudo and realistic game traffic to transfer, that is, same simulation was repeated twice with the pseudo and realistic game traffic, respectively. The pseudo game traffic consisted of randomly selected (generated) samples by the analytical model, the realistic game traffic was generated by replaying the original empirical game traffic trace. Game traffic was collected in its network sink and the metric of inter-arrival time was parsed.

In the data analysis, Mann-Whitney test was performed with random samples of the pseudo game traffic by the analytical model against those of the realistic game traffic. The test results determined the applicability of the analytical game traffic model.

CHAPTER 3

RESULTS

3.1. RESULTS OF TEST-BED EXPERIMENT

The test-bed experiment was performed with up to five game players. The empirical game traffic was collected in the experiment, the data of different traffic metrics were parsed, stored and analyzed per client. Five sets of empirical data were obtained for one traffic metric because there were five clients in the game.

The game traffic metrics in this research was inter-send time and packet size. The collected empirical traffic consisted of server traffic and client traffic. The analysis of server traffic and client traffic was performed separately, therefore there were four traffic metrics and they were the server inter-send time, the client inter-send time, the server packet size, and the client packet size.

The game Quake offers different maps for selection. In order to investigate different game traffic behaviour among different maps, the test-bed experiment was repeated three times (sessions) with different maps. The numbers of clients in these three sessions of the experiment were four, four, and five, respectively. There were in total thirteen clients in the test-bed experiment.

The presentation of the results of the test-bed experiment is organized in this way: data analysis of the same traffic metric of all (thirteen) clients were presented in same texts, data of the same metric of all clients were shown in same table, and plots of the same metric of all clients were usually drawn in same figure.

3.1.1. INTER-SEND VERSUS INTER-ARRIVAL TIME

The inter-packet time is the time between two consequent packets. It can be inter-send time (at traffic source) or inter-arrival time (at traffic sink) if traffic collection location is concerned. In the test-bed experiment, the empirical data of the metrics of both

inter-send time and inter-arrival time were obtained. After careful examination of plots of both metrics, there was no difference between the inter-send time and inter-arrival time. For example, one plot of the server inter-send time is shown in Figure 3, its corresponding server inter-arrival time plot is shown in Figure 4.

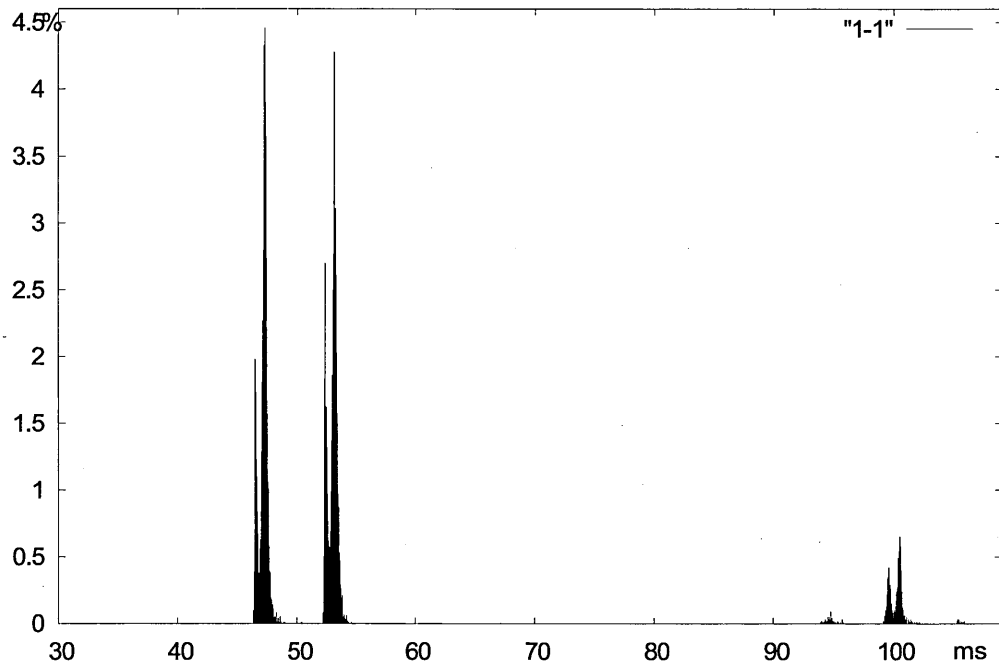


Figure 3 Server inter-send time in test-bed experiment

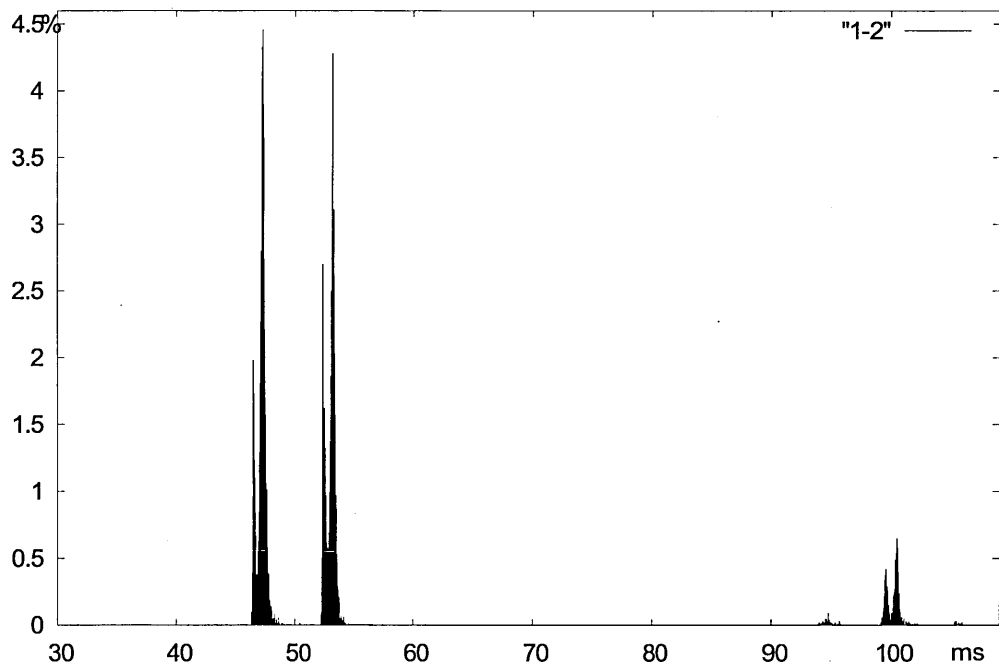


Figure 4 Server inter-arrival time in test-bed experiment

Inter-arrival time had the same characteristics of inter-send time in case that there was no other traffic and network bandwidth was abundant. The test-bed experiment offered the “clean” game traffic behaviour without other traffic, and inter-packet time was same at both sending and arrival locations. In order to distinguish the “clean” traffic in the test-bed experiment from the “interfered” traffic in the Internet experiment, the term “inter-send time” was used for the inter-packet time in the test-bed experiment.

3.1.2. CLIENT INTER-SEND TIME

3.1.2.1. CLIENT INTER-SEND TIME OVERVIEW

The empirical data of the client inter-send time were obtained, all thirteen plots of the client inter-send time were drawn together in the next figure. The thirteen plots not only had same curve-shape, but were also highly overlapped one another. One primary characteristic of the client inter-send time plots was a narrow peak, and the peak was at 33 ms for all thirteen plots.

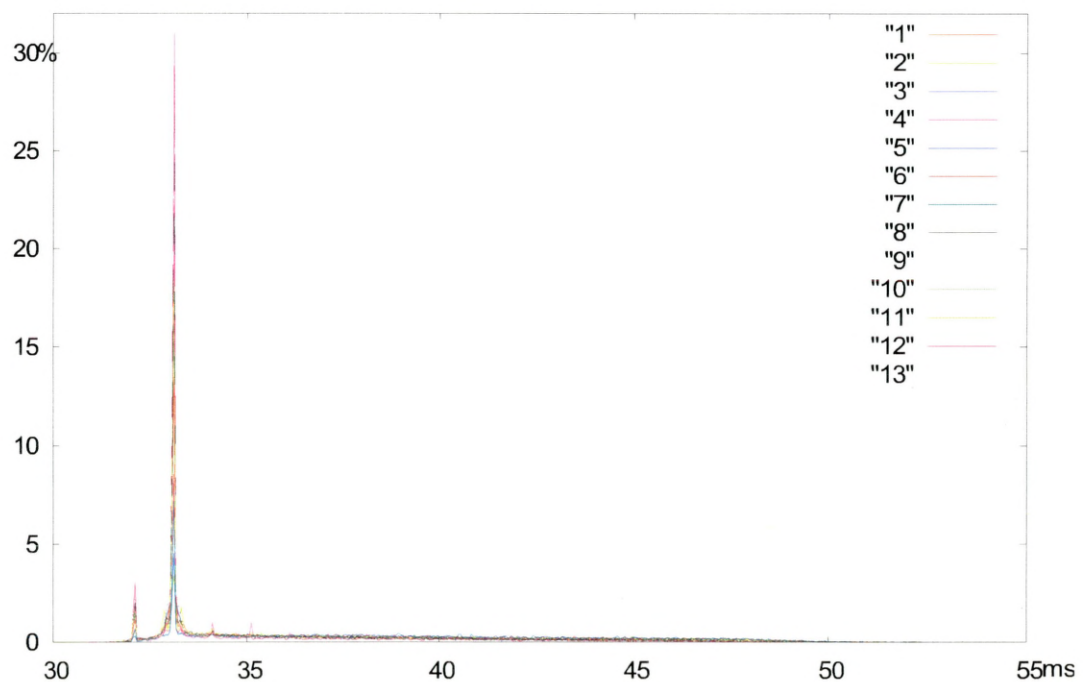


Figure 5 Client inter-send time (all clients)

3.1.2.2. DETERMINISTIC DISTRIBUTION

The deterministic (or degenerate) distribution is the probability distribution of a discrete random variable that assigns all of the probability, that is, probability 1, to a single number, a single point, or otherwise to just one outcome of a random experiment.

The probability density function of deterministic distribution is:

$$f(x) = \begin{cases} 1, & \text{if } x = c \\ 0, & \text{if } x \neq c \end{cases} \quad (4)$$

where c is a constant.

Deterministic distribution is localized at a point c in its plot. The plot of the deterministic probability distribution is featured by a single peak.

3.1.2.3. SUPPLEMENT

The client inter-send time approximately had deterministic distribution, as shown in Figure 5. However, the peak at 33 ms did not contain the inter-send time of all client packets, some empirical data of the client inter-send time were distributed between 31 and 55 ms, as shown in the next figure (it is an amplified figure of a part of Figure 5).

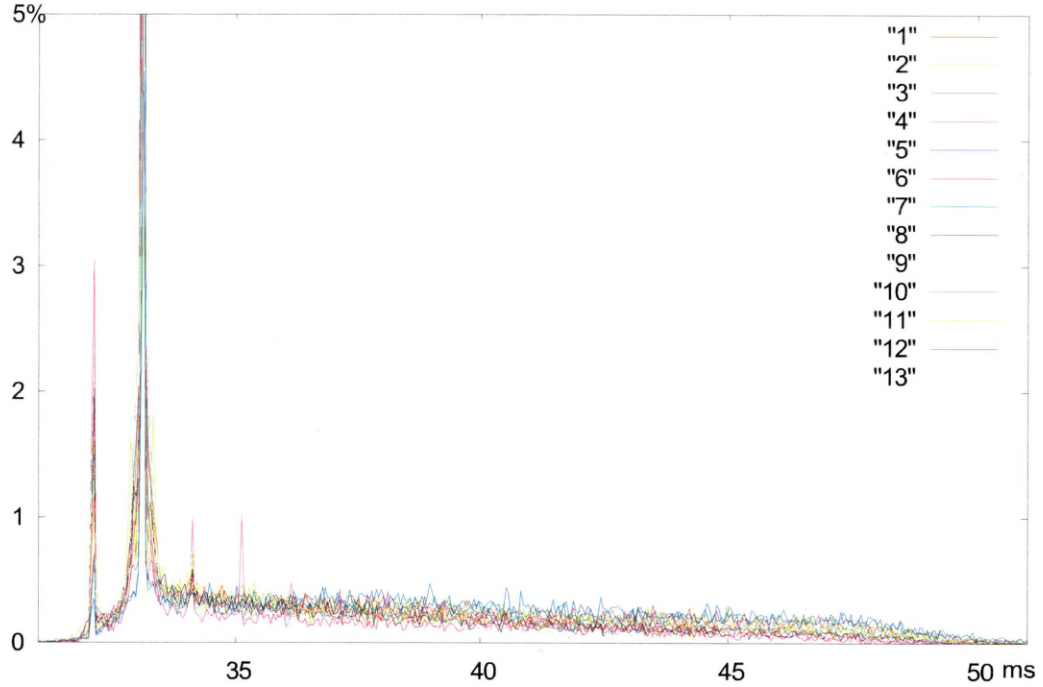


Figure 6 Client inter-send time (amplified)

The empirical data of the client inter-send time were statistically calculated, 35 % of total inter-send time was 33 ms (exactly, between 32.65 and 33.65 ms), 65% of total inter-send time was roughly uniformly distributed between 31 and 49 ms (except 33 ms). The client inter-send time approximately had the dual-distribution of the deterministic and the uniform. Different clients had the same characteristics of inter-send time.

3.1.3. SERVER INTER-SEND TIME

3.1.3.1. SERVER INTER-SEND TIME PLOTS

The server inter-send time approximately had deterministic distribution, as shown in the next figure. All thirteen plots had two major peaks which were at 47 ms and 53 ms, respectively, and some plots had a minor peak at about 100 ms.

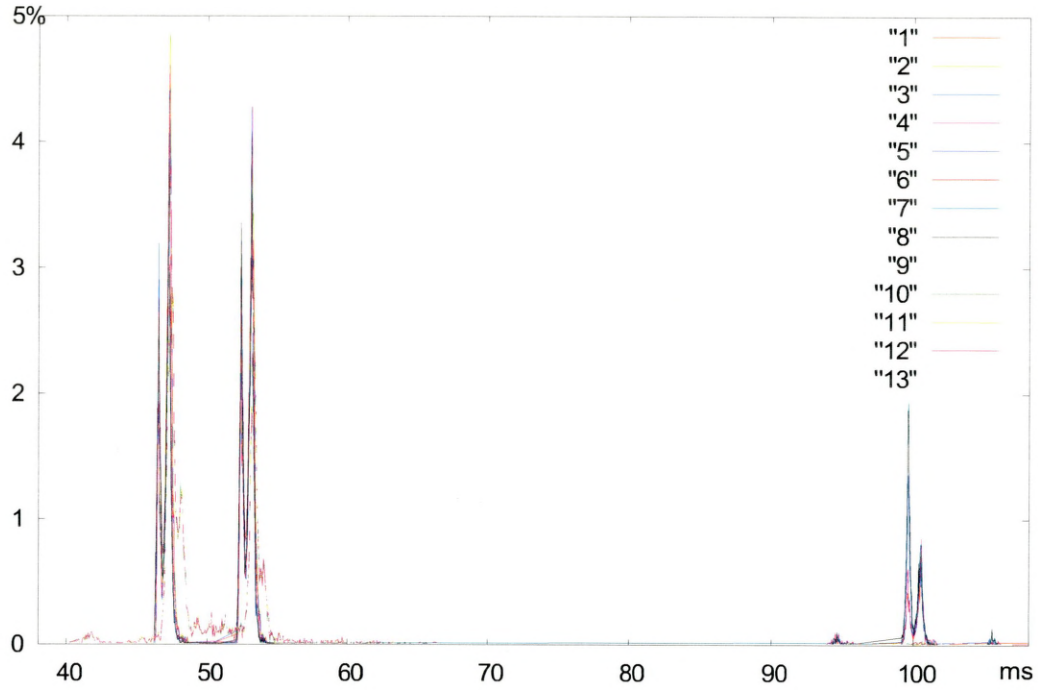


Figure 7 Server inter-send time to all clients

3.1.3.2. SERVER INTER-SEND TIME DATA

The empirical data of the server inter-send time to every client are shown in the next table. The data to every client within one game session are organized in the consecutive rows. The column is organized by empirical data around every peak. The first column in every peak is the peak location (in micro second, or μs) and the second column is the percentage of inter-send time in the 100- μs interval centred the peak. The server inter-send time was deterministic distributed because 46.7 % of the inter-send time was 47 ms, 40.3% of the inter-send time was 53 ms, 5.6% of the inter-send time was approximately 100 ms. The rest inter-send time was only 7.4%, and the 7.4% of the inter-send time was primarily centred slightly further away (beyond the 100- μs interval) from these two major peaks, due to careful examination of the empirical data.

Session	Client	Peak 1		Peak 2		Peak 3	
		Location(μ s)	%	Location(μ s)	%	Location(μ s)	%
1	1	47275	45.9	53125	43.6	100575	7.9
	2	47275	50.9	53125	48.1	none**	0
	3	47275	45.1	53125	43.1	100525	8.5
	4	47275	44.7	53125	42.9	100575	8.9
2	1	47225	44.1	53075	43.5	99625	12.6
	2	47225	48.8	53125	50.4	none**	0
	3	47225	38.9	53125	41.3	99625	17.0
	4	47225	39.3	53075	41.5	99575	16.6
3	1	47375	49.9	53175	34.2	uniform*	0.4
	2	47375	49.9	53225	34.3	uniform*	0.3
	3	47375	50.0	53175	33.4	uniform*	0.3
	4	47375	50.0	53175	33.1	uniform*	0.3
	5	47375	50.1	53175	34.4	uniform*	0.3
Average			46.7		40.3		5.6

Table 4 Percentage at 1-ms interval at peaks of server inter-send time

* *Uniform: the empirical data (in Session 3) were roughly uniformly distributed in the interval approximately between 80 ms and 120 ms.*

***None: No packet in the empirical data had inter-send time around 100 ms.*

The server inter-send time was primarily same (for example, the percentage on both major peaks) in all three sessions, but was slightly session (map selection in server) dependent (for example, the inter-send time around 100 ms was a minor peak in Session 1 and Session 2, but was uniformly distributed in Session 3).

3.1.4. CLIENT PACKET SIZE

The distribution of the client packet size was simple, as shown in the next figure. The thirteen plots of the client packet size highly overlap one another, and every impulse in the figure is the highest impulse among these thirteen impulses.

The range of the client packet size distribution was small, because the client packet size was between 37 and 64 bytes. The packet size of 43-byte had more packets than other packet size. The average percentage at every packet size is shown in the next table (the total percentage from all 28 cells is 100%).

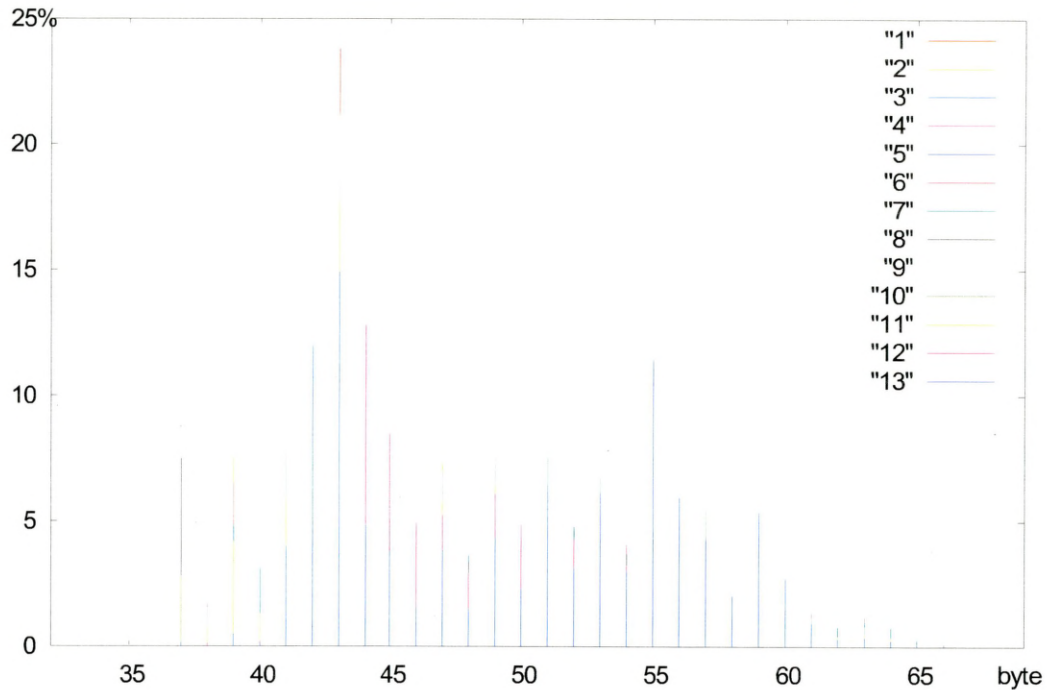


Figure 8 Client packet size distribution

Size(byte)	37	38	39	40	41	42	43	44	45	46	47	48	49	50
Percentage	1.3	0.7	3.2	1.0	6.4	3	15.9	5	6.7	2.7	5.8	2.1	6.3	2.5
Size(byte)	51	52	53	54	55	56	57	58	59	60	61	62	63	64
Percentage	6.8	2.8	5.6	2.2	6.0	2.2	3.7	1.3	3.5	1.2	0.9	0.4	0.6	0.2

Table 5 Percentage of client packet size out of total packets

The client packet size was primarily distributed in the interval between 39 and 59 bytes (the client packet size was fully distributed in the interval between 37 and 64 bytes). The x-axis location of 43-byte formed a minor peak compared to all other packets sizes because it had 15.9% of total packets.

3.1.5. SERVER PACKET SIZE

3.1.5.1. SERVER PACKET SIZE OVERVIEW

The empirical data of the server packet size were more complex than those of the above three metrics. The plots of the server packet size from different game sessions were slightly different and therefore drawn separately (plots to all clients in one session were drawn in one figure), as shown in Figure 9, 10 and 11, respectively.

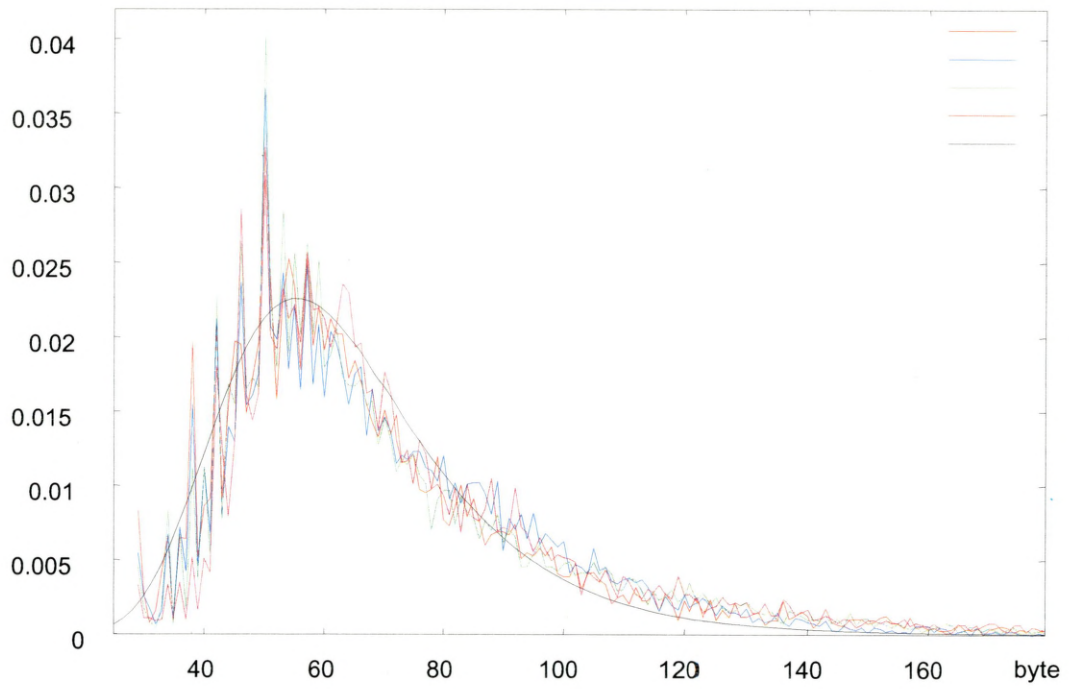


Figure 9 Server packet size in test best experiment, session 1 (to 4 clients)

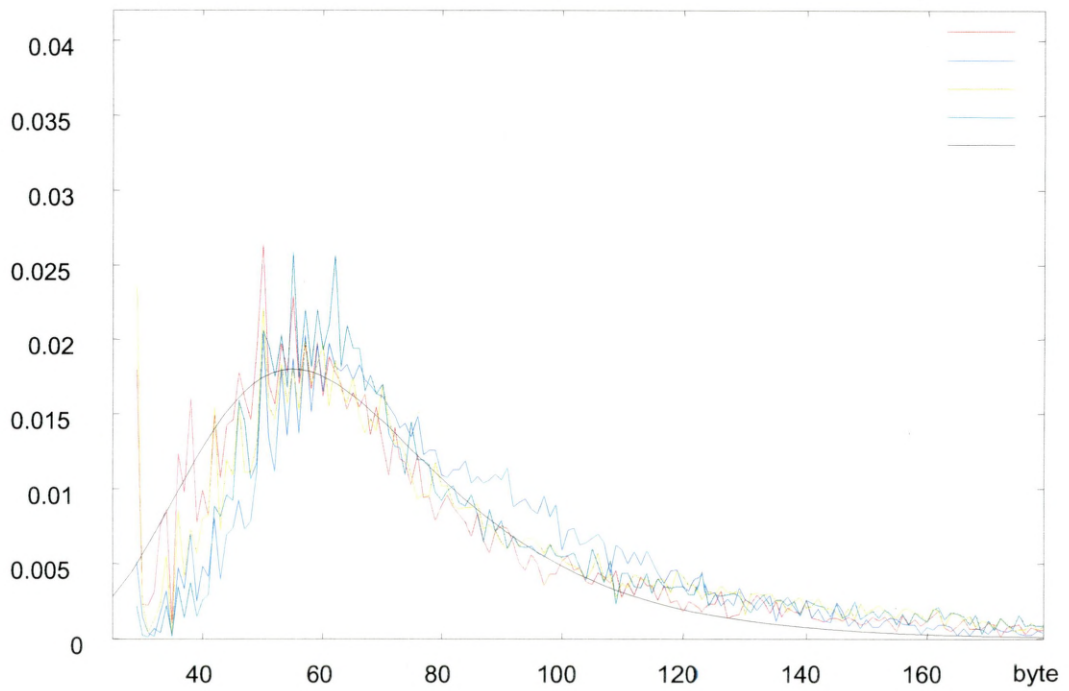


Figure 10 Server packet size in test best experiment, session 2 (to 4 clients)

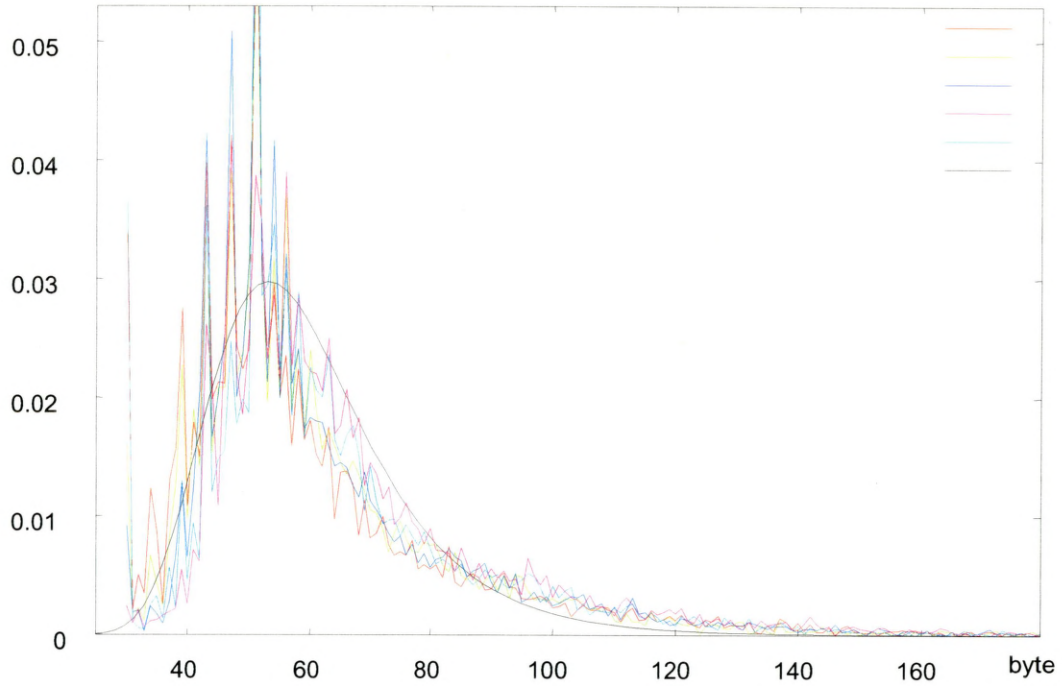


Figure 11 Server packet size in test best experiment, session 3 (to 5 clients)

The variation among the plots in same figure was minor compared to the variation among the plots in different figures. Therefore the difference of the server packet size to different clients was minor, and the server packet size was session (map) dependent. However, the main characteristics (for example, the range of the packet size distribution, the general shape of plots) were same among different sessions. Therefore it was possible to get a same analytical distribution for all empirical server packet sizes in different game sessions.

3.1.5.2. EXTREME DISTRIBUTION

The general formula for the probability density function of the extreme probability distribution is:

$$f(x) = \frac{1}{\beta} e^{-\frac{x-\mu}{\beta}} e^{-e^{-\frac{x-\mu}{\beta}}} \quad (5)$$

The cumulative distribution function is:

$$F(x) = e^{-e^{\frac{x-\mu}{\beta}}} \quad (6)$$

where μ is the location parameter and β is the scale parameter.

3.1.5.3. PARAMETERS

The server packet size plots in the above three figures look like extreme distribution. They were drawn together with a plot of extreme distribution. The plot of extreme distribution in every figure was the analytical fitting curve to one empirical plot in the same figure (the analytical plot fitted to the plot of Client 3 in Figure 9, to Client 1 in Figure 10, to Client 5 in Figure 11). The curve fitting was performed by the program Gnuplot.

Any extreme distribution is determined by two parameters, and they are location parameter μ and scale parameter β . The location and scale parameters of the thirteen plots were obtained during curving fitting and shown in the next table. The average location parameter was 55.6, the average scale parameter was 16.2.

Session	Client	μ	β
1	1	55.2	16.8
	2	56.5	18.1
	3	55.3	16.2
	4	57.7	16.4
	average	56.2	16.9
2	1	55.4	20.0
	2	65.0	21.4
	3	59.6	21.7
	4	60.6	18.2
	average	60.2	20.3
3	1	48.5	13.1
	2	50.6	12.4
	3	50.8	11.0
	4	54.1	12.5
	5	53.1	12.4
	average	51.4	12.3
Minimum		48.5	11.0
Maximum		65.0	21.7
Average		55.6	16.2

Table 6 Location and scale parameters of extreme distribution

Although Gnuplot curve fitting indicated that the server packet size plots fitted extreme distribution, the error estimation in curve fitting results from Gnuplot is over-optimistic (according to Gnuplot's user manual). Therefore some other curve fitting was required.

3.1.5.4. CURVE FITTING

A specific curve fitting method was performed to the server packet size curves. The method was the m-trial K-S test. The m-trial K-S test was to check the number of total rejects after K-S test was repeated m times. The curve fitting was determined by the number of rejects in the m trials of K-S test. The m-trial test was 30-trial test in this section.

The sample-size was fixed in the m-trial test. If the sample-size was set to 100, the K-S test was performed with 100 random samples. The same K-S test was repeated m (30 in this section) times and the 100 random samples were different every time. The 30-trial test with the sample-size 100 was performed to every of the thirteen plots, the test results were the numbers of non-rejects in these 30-trial tests. The results are shown in Column 100 (data in thirteen rows were server packet sizes to the thirteen clients) in Table 7.

Session	Client	Sample-size									
		36	49	64	81	100	121	144	169	196	225
1	1	28	28	25	30	26	27	25	28	26	24
	2	26	28	27	27	27	27	25	28	26	22
	3	29	29	25	25	26	26	22	21	24	17
	4	23	28	28	24	26	27	26	23	20	23
2	1	25	28	27	21	22	20	19	27	20	23
	2	27	30	27	29	27	27	27	27	28	27
	3	28	28	25	25	19	25	24	21	22	19
	4	24	27	28	25	21	26	25	20	14	18
3	1	26	26	27	26	23	24	24	23	21	20
	2	26	26	27	27	23	24	28	26	24	26
	3	28	26	29	24	24	21	20	21	18	18
	4	28	25	26	23	26	19	22	23	15	20
	5	26	28	27	27	24	26	25	21	23	16

Table 7 Server packet size test results

The 30-trial test was repeated with nine other sample-sizes after the sample-size 100. The nine other sample-sizes were 36, 49, 64, 81, 121, 144, 169, 196 and 225. The

results (number of non-rejects) of these 30-trial tests are shown in Column 36, 49, 64, 81, 121, 144, 169, 196 and 225 in Table 7. The cells with the number of non-rejects between 25 and 30 (inclusive) are shaded. The shaded cells indicate that the null hypothesis of extreme distribution should not be rejected.

Table 7 shows the whole results of the above-described test with ten different sample-sizes (36, 49, 64, 81, 100, 121, 144, 169, 196 and 225). Because the results (shaded cells) were not regular distributed in Table 7, the above-described test with ten different sample-sizes was performed once again and results are shown in Table 8.

Session	Client	Sample-size									
		36	49	64	81	100	121	144	169	196	225
1	1	27	28	30	27	26	25	24	27	28	24
	2	27	28	27	30	27	29	27	27	28	26
	3	26	27	22	23	26	29	27	25	22	20
	4	30	27	27	26	23	25	21	22	21	23
2	1	28	26	27	27	27	26	25	23	24	19
	2	26	29	27	27	28	28	29	28	26	26
	3	27	27	29	27	26	26	25	22	21	22
	4	25	27	27	25	21	20	19	10	12	18
3	1	28	25	27	23	25	26	21	27	21	20
	2	26	28	26	26	26	22	26	21	28	24
	3	28	27	26	19	24	21	19	20	17	16
	4	26	22	23	20	27	26	22	21	22	18
	5	27	28	23	25	28	25	25	23	23	20

Table 8 Repeated server packet size test results

In the above two tables, most cells in the columns of small sample-size (Column 36, 49, and 64) are shaded (non-rejected) cells, they support the null hypothesis of extreme distribution. And there are many non-rejected cells in the columns of other sample-sizes (from Column 81 to Column 225) in both tables. However, there are many rejected cells in both tables (especially from Column 81 to Column 225), the rejected cells were often caused by outliers, after careful examination of the details of the 30-trial test. The proportion of the number of packets in outliers was small compared to the total number of packets.

The null hypothesis of extreme distribution was more likely to be accepted with small sample-sizes (36, 49 and 64); if larger sample-sizes (81, 100, 121, 144, 169, 196, and 225) were considered, the server packet size to only a few clients (for example, Client 1 and Client 2 in Session 1, Client 2 in Session 2) had extreme distribution.

The results in each table were from a whole procedure of the above-described test. The whole procedure of the above-described test was repeated eight more times (ten times including the above two), the average numbers of non-rejects from all ten whole procedures are shown in the next table. The shaded cells indicate the tendency of supporting extreme distribution.

Sample-size	49	64	81	100	121	144	169	196
Average non-reject	28.2	27.3	26.1	24.9	23.7	22.7	21.8	19.5

Table 9 Average server packet size test results

Apart from the sample-sizes of 36, 49, 64, 81, 100, 121, 144, 169, 196, and 225, the 30-trial K-S test was also performed with smaller sample-sizes (less than 36), and the results had larger number of shaded cells than the results in Table 7 and 8.

Based on the above 30-trial K-S test results, the server packet size approximately fitted to extreme distribution.

3.1.6. BASIC STATISTICS

The basic statistics (minimum, maximum, and average) of packet size, inter-send time, baud rate and packet rate of the test-bed experiment were obtained and are shown in the next table (discussions on the next table are in Chapter Four).

	Packet size (bytes)		Inter-send time (ms)		Data rate (Kbps)		Number of packets (per second)	
	Server	Client	Server	Client	Server	Client	Server	Client
Minimum	29	33	29	28				
Maximum	457	68	199	735				
Average	73.3	48.2	52.9	37.1				

Table 10 Basic statistics of traffic in test-bed experiment

3.2. RESULTS OF INTERNET EXPERIMENT

In the Internet experiment, the same experiment was repeated many times (sessions). The number of total players was determined by server selection and was about five in sessions of non-team game type (Free-for-All) and was about twelve in sessions of

team game types (Team-Deathmatch or Catch-the-Flag) of the Quake III Arena. One client was unique in the Internet experiment because this client was in the University of Abertay Dundee but the server and all other clients in same experiment session were in other locations on the Internet (server and other clients were different in different experiment sessions), and because the empirical traffic of the Internet experiment was collected in this client's host computer.

After both inbound and outbound traffic in this client's host computer was collected, the empirical data of traffic metrics were parsed and metric plots were drawn. According to this experiment design, the metrics of inter-packet time in the Internet experiment were the client inter-send time and the server inter-arrival time.

3.2.1. REGULAR AND IRREGULAR PLOTS

Although the plots of the same metric in the test-bed experiment were similar, there were differences among the different plots of the same metric in the Internet experiment. Some plots of the same metric in the Internet experiment even had different shapes, although most plots of the same metric were similar. A small percentage of plots were irregular compared to most other plots of the same metric in the Internet experiment. These irregular or different plots were ignored and not presented in this dissertation. Plots in this section represent the general distribution of game traffic in the Internet experiment.

3.2.2. INTER-PACKET TIME

3.2.2.1. CLIENT INTER-SEND TIME

The plots of the client inter-send time distribution in the Internet experiment were drawn. They were compared to the client inter-send time plot of the test-bed experiment. The plot in Figure 12 is the client inter-send time in the test-bed experiment, the plots in Figure 13 and 14 are the client inter-send time in the Internet experiment.

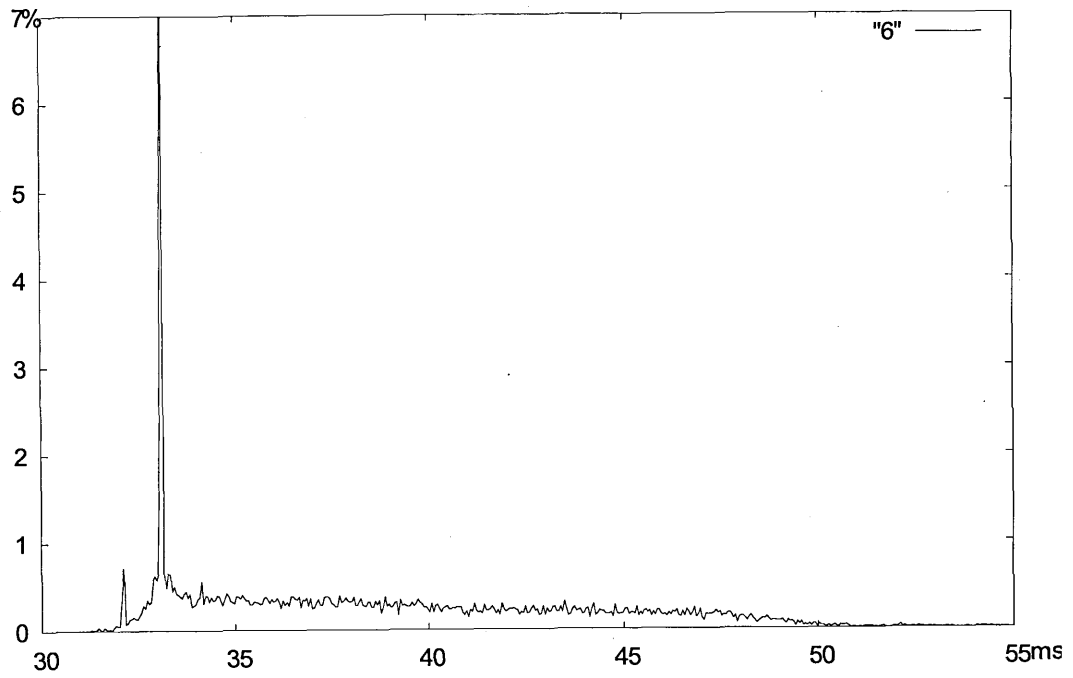


Figure 12 Client inter-send time in test-bed experiment

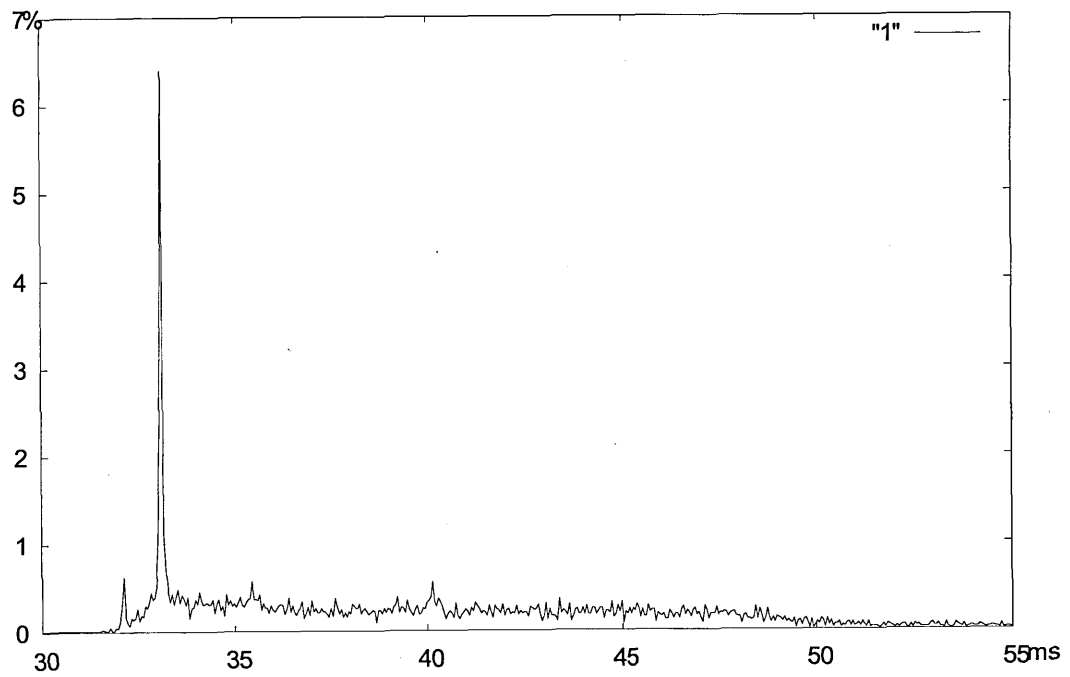


Figure 13 Client inter-send time in Internet experiment, session 1

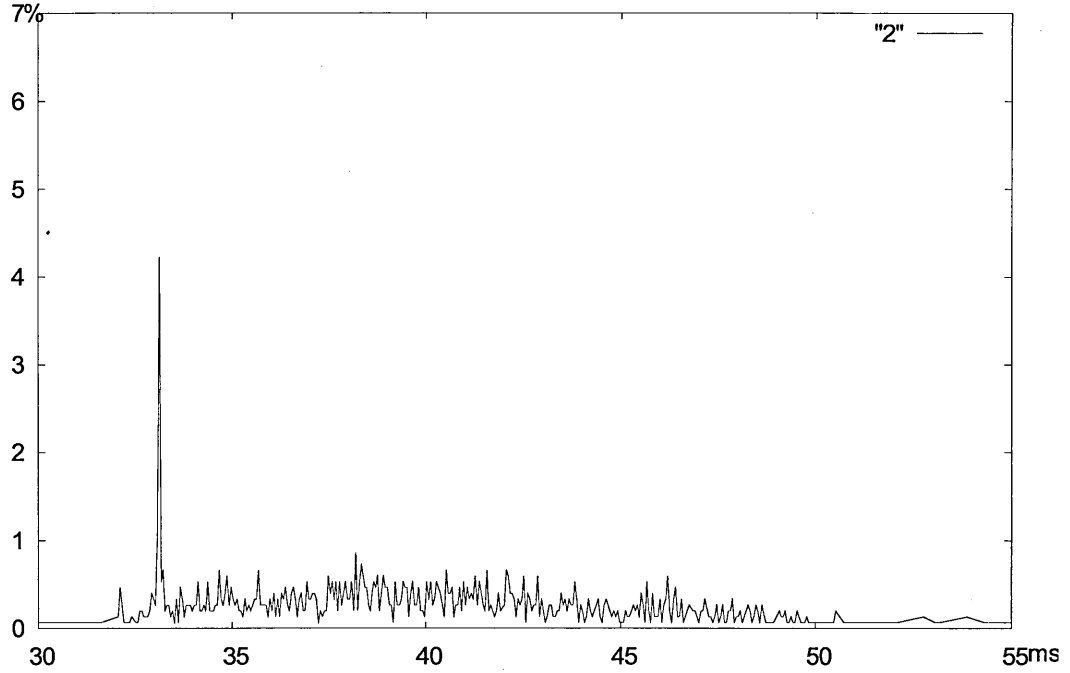


Figure 14 Client inter-send time in Internet experiment, session 2

In the Internet experiment, many client packets were sent at the fixed interval of 33 ms. Many other client packets were sent at the interval between 31 and 50 ms (except 33 ms). The plots in of the Internet experiment had the same shape as the plot of the test-bed experiment, therefore the Internet experiment approximately had the same distribution of the client inter-send time as the test-bed experiment. However, the primary variation of the Internet traffic from the test-bed traffic was that the peak (33 ms) in the test-bed plot was higher than that in the Internet plot. Therefore not so many client packets were transmitted at the fixed interval of 33 ms in the Internet experiment as in the test-bed experiment.

Similar to the test-bed experiment, the client inter-send time in the Internet experiment approximately had the dual-distribution of the deterministic and the uniform.

3.2.2.2. SERVER INTER-ARRIVAL TIME

The empirical data of the server inter-arrival time were obtained in the Internet experiment and the server inter-arrival time plots were drawn. The characteristics of this metric were shown in these plots, and insights were obtained by the comparison

between the plots of the test-bed and the Internet experiments, or among plots of the Internet experiment themselves. The plot in Figure 15 is the server inter-arrival time in the test-bed experiment, the plots in Figure 16 to 24 are the server inter-arrival time in the Internet experiment. The plots in Figures 16 to 21 are based on non-team type (Free-for-All), whereas the plots in Figures 22 to 24 are based on team types (Team-Deathmatch or Catch-the-Flag).

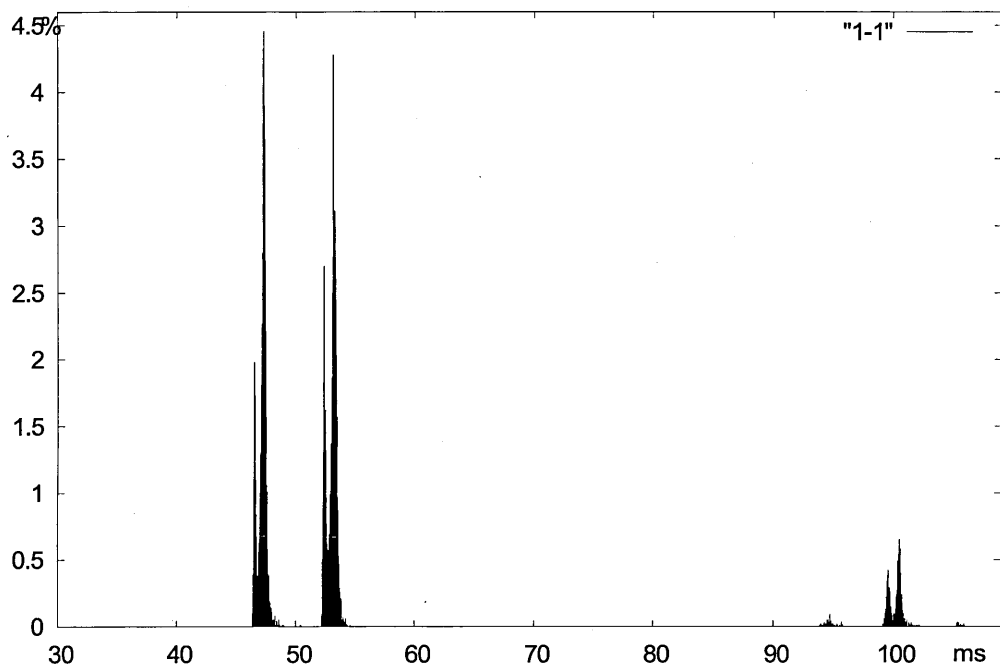


Figure 15 Server inter-arrival time in test-bed experiment

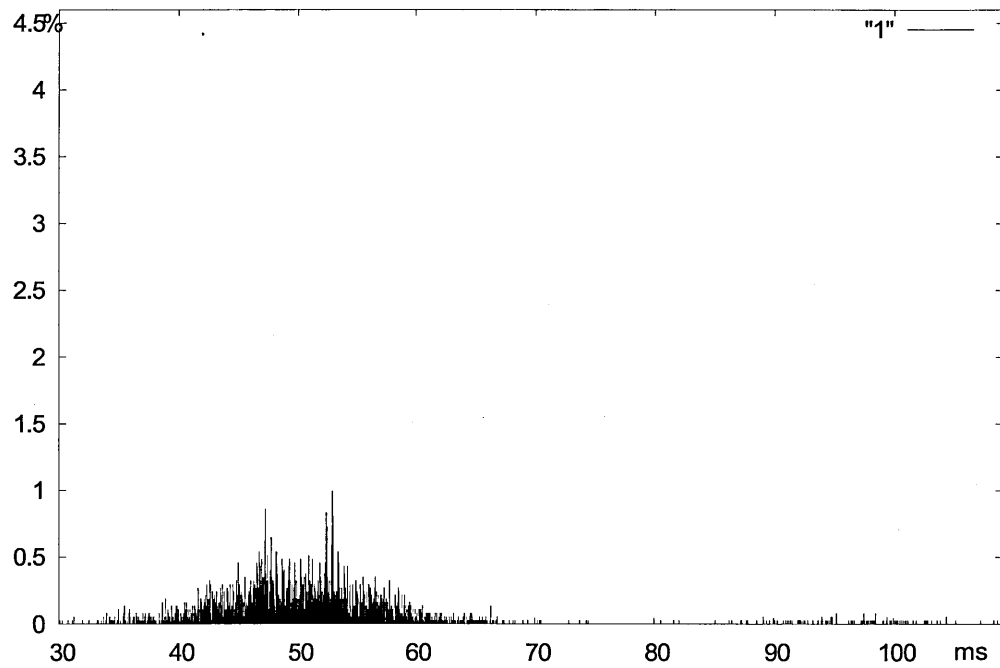


Figure 16 Server inter-arrival time in Internet experiment, Session 1

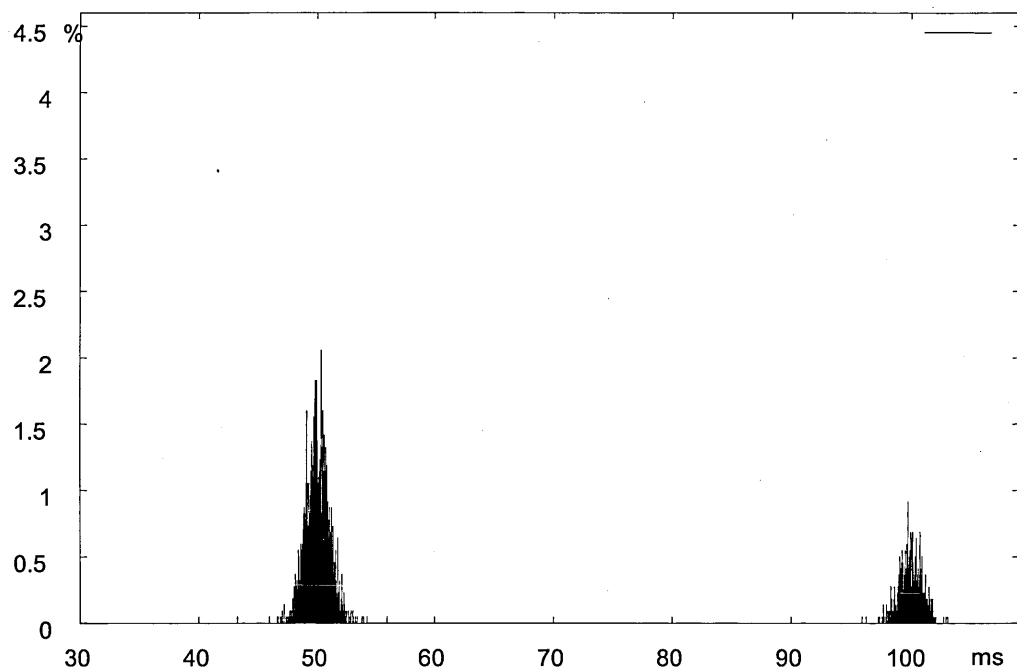


Figure 17 Server inter-arrival time in Internet experiment, Session 2

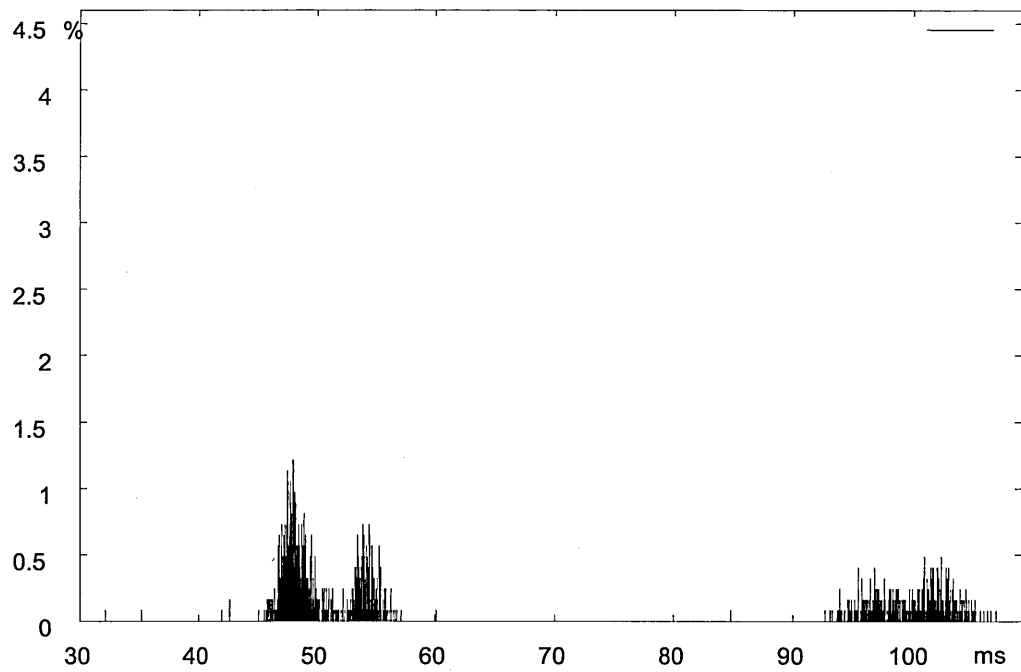


Figure 18 Server inter-arrival time in Internet experiment, Session 3

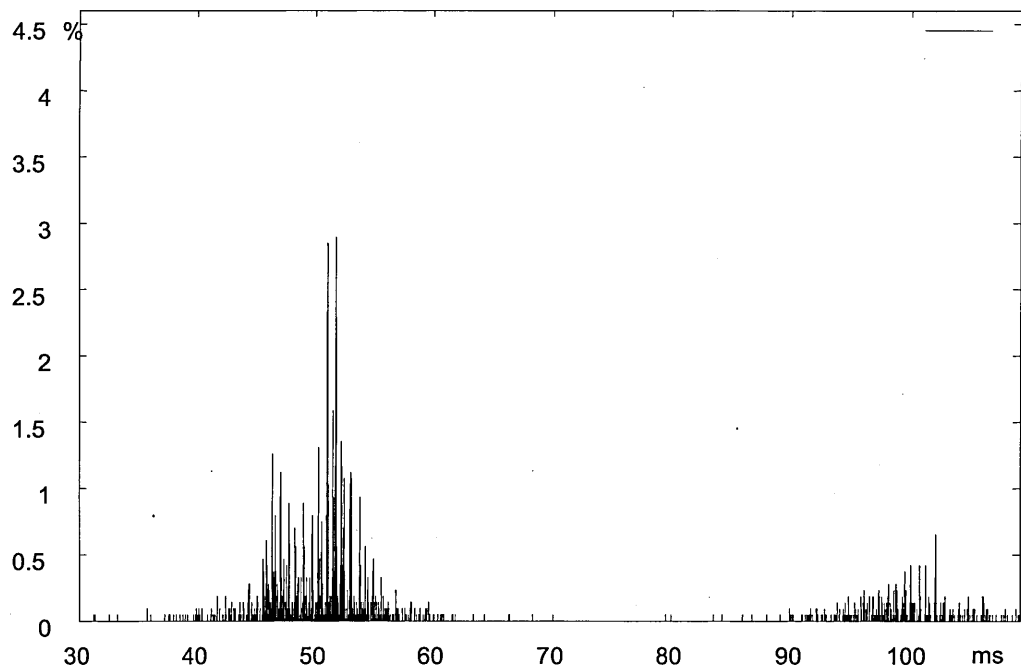


Figure 19 Server inter-arrival time in Internet experiment, Session 4

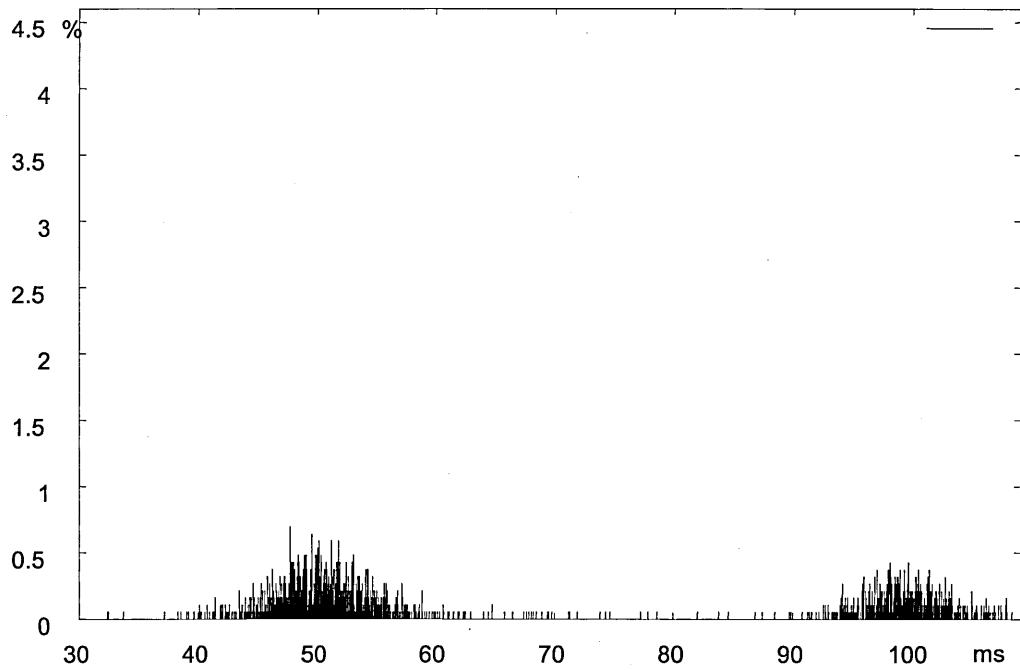


Figure 20 Server inter-arrival time in Internet experiment, Session 5

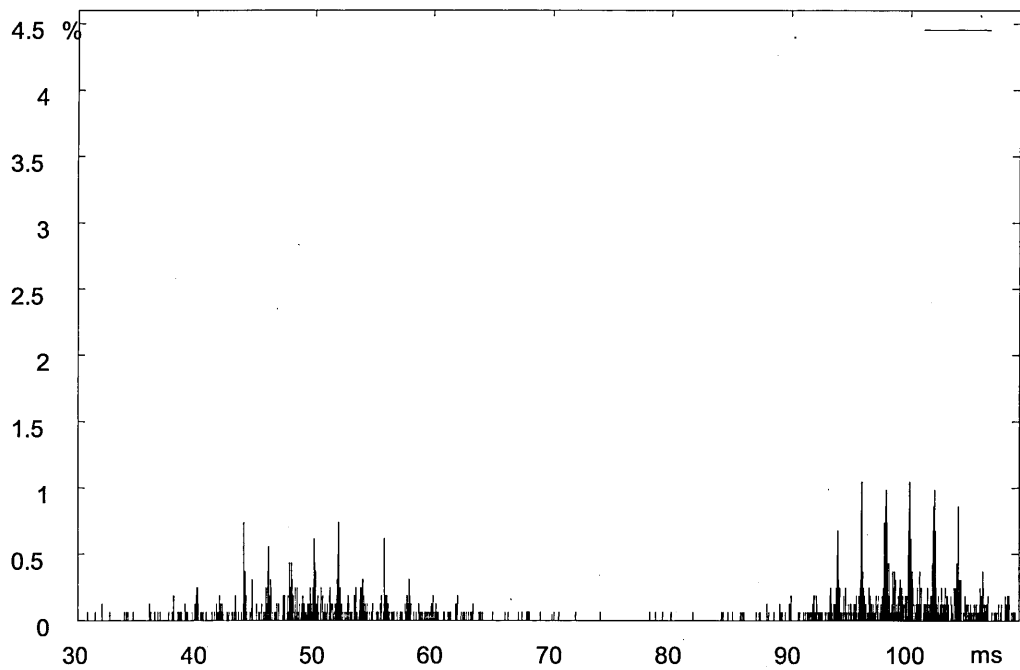


Figure 21 Server inter-arrival time in Internet experiment, Session 6

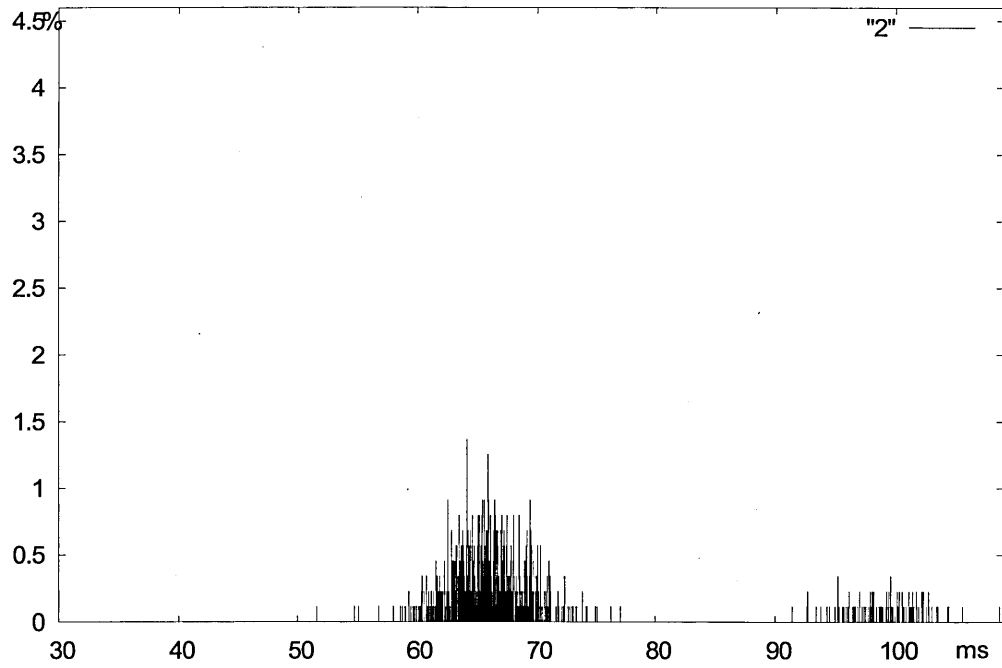


Figure 22 Server inter-arrival time in Internet experiment, Session 7

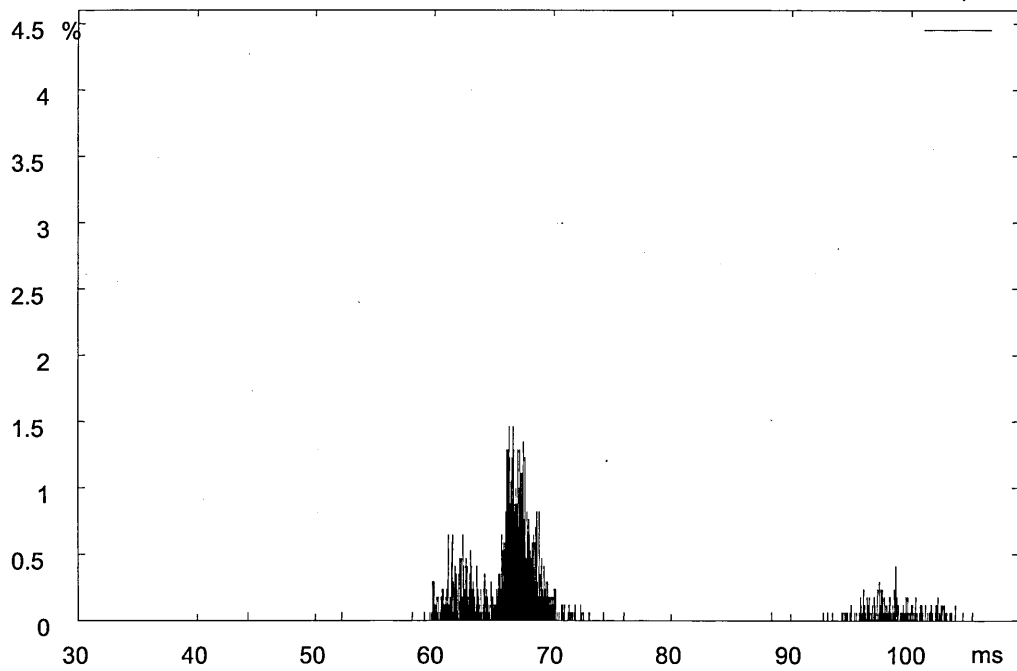


Figure 23 Server inter-arrival time in Internet experiment, Session 8

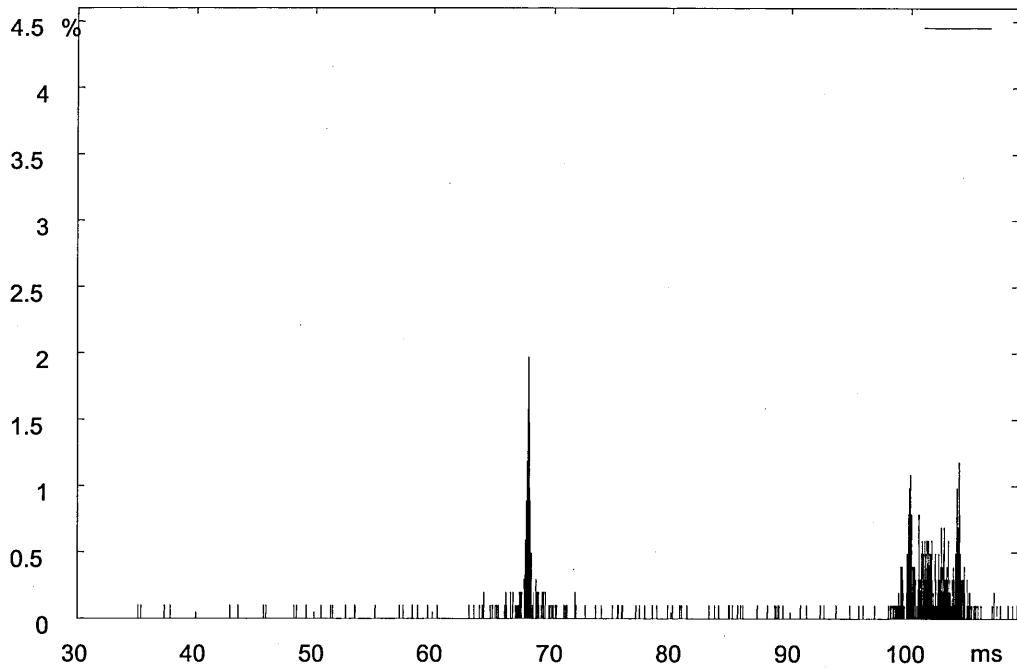


Figure 24 Server inter-arrival time in Internet experiment, Session 9

The game server-client delay was different in different sessions of the Internet experiment. The server-client delay in Figure 16 to 19 was less than 50 ms, whereas the server-client delay in Figures 20 and 21 were 83 and 122 ms, respectively. The server inter-arrival time from farther away servers usually had larger cluster (higher percentage) around 100 ms. Game server sent less packets to farther away clients, and/or more packets were dropped during network transfer from a farther away server.

The plots of the server inter-arrival time of non-team (Figure 16 to 21) and team (Figure 22 to 24) game types were different. The major difference was that team types had longer inter-arrival time (around 70 ms) than non-team type (around 50 ms).

The server inter-arrival time plots in the Internet experiment were irregular compared to the server inter-send time plot in the test-bed experiment. However, the Internet empirical plots preserved characteristics of the test-bed empirical plots to a certain extent. The server inter-send time in the test-bed was featured by the sharp dual-peak (the two major peaks), the dual-peak was replaced by a bell-shaped cluster in the server inter-arrival time plots in the Internet experiment. And the minor peak in the test-bed plot can still find its corresponding cluster in the same x-axis location of the plots in the Internet experiment. The peaks in the plot of the text-bed experiment were

transformed and became the clusters in the plots of the Internet experiment (this situation is further discussed in the simulator experiment).

3.2.3. CLIENT PACKET SIZE

A plot of the client packet size in the Internet experiment is shown in next figure. Compared to the client packet size distribution in Section 3.1.4, the client packet size in the Internet experiment did not have big difference from that in the test-bed experiment. Other traffic in the network did not cause big change of the client packet size.

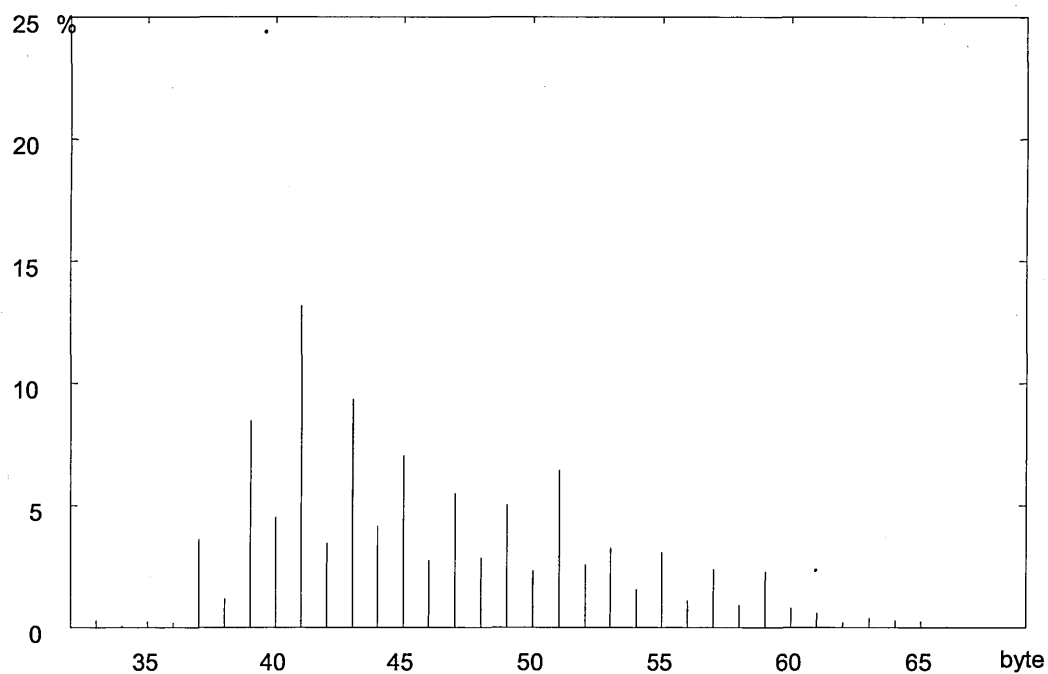


Figure 25 Client packet size in Internet experiment

3.2.4. SERVER PACKET SIZE

A plot of the server packet size in the Internet experiment is shown in next figure.

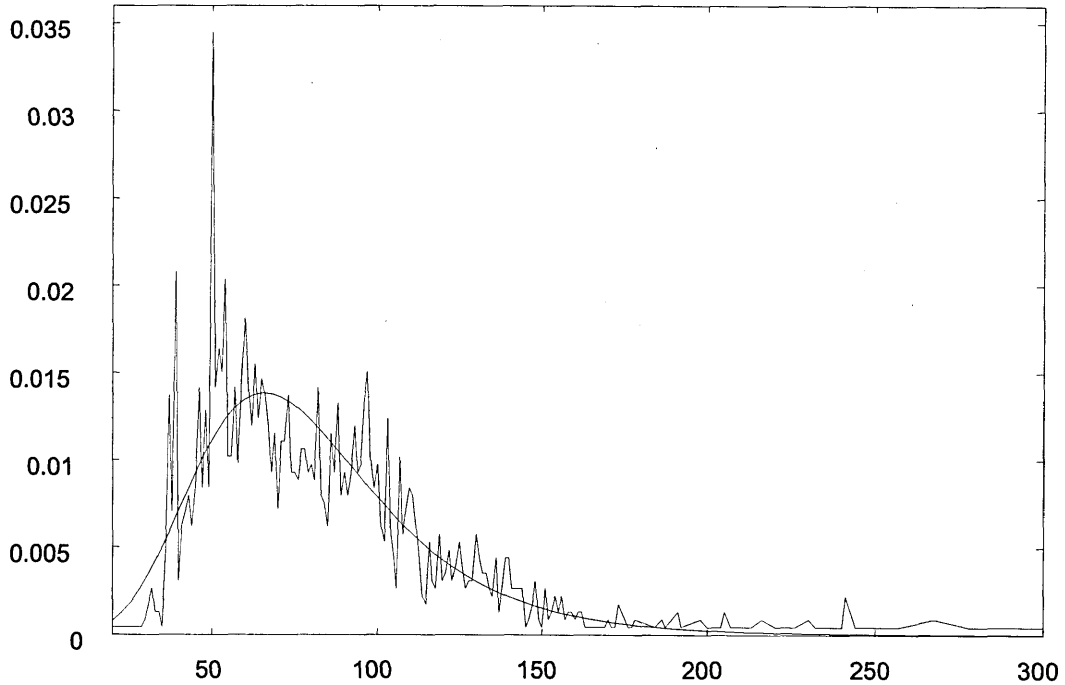


Figure 26 Server packet size in Internet experiment

There is an approximate fitting curve of extreme distribution generated by Gnuplot in the above figure. The server packet size is highly variable compared to the approximate fitting curve. The location parameter and scale parameter of the approximate fitting curve were 66 and 27, respectively. Both parameters are near the average location and scale parameters (56 and 16, respectively) in the test-bed experiment. The server packet size in the Internet experiment did not have very big difference from that in the test-bed experiment, although server packet size in the Internet experiment is highly variable. Other traffic in the network did not cause big change of the server packet size.

3.3. RESULTS OF SIMULATOR EXPERIMENT

Three simulator experiments were performed and results were interpreted. The first simulation showed the progression from the “clean” to the “interfered” game traffic, the second compared the effects on game traffic from different types of traffic, and the third evaluated the analytical game traffic model.

3.3.1. PROGRESSION FROM “CLEAN” TO “INTERFERED” TRAFFIC

The objective of this simulator experiment was to observe the game traffic progression from inter-send time to inter-arrival time with other traffic. The result of this experiment successfully showed the server inter-packet time progression from the dual-peak distribution to the bell-shaped distribution.

This experiment was performed by ns2 simulator with the network configuration in Figure 2 in Chapter Two. Game traffic in the simulation was generated by replaying the collected game traffic in the test-bed experiment, the other traffic was a FTP stream in each direction of the shared link. Different parameters of the queue length (queue length was set to 16 after many tries) and bandwidth of the shared link were tried in order to obtained the expected progression.

The server inter-send time was deterministic distributed and is shown Figure 3 in Section 3.1.1. The empirical data of the server inter-arrival time with impact from the FTP streams were collected five times, when the bandwidth of the shared link was 2.5, 2.0, 1.5, 1.0, and 0.5 MB, respectively. The five plots of the server inter-arrival time of simulation are shown in the next five figures.

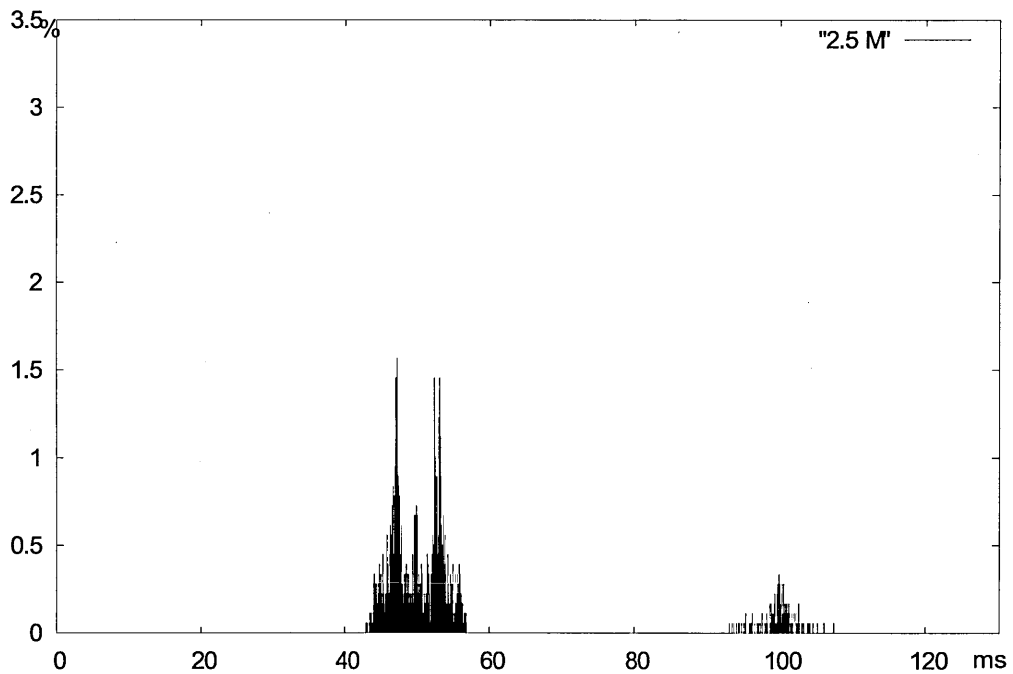


Figure 27 Server inter-arrival time with bandwidth 2.5 MB

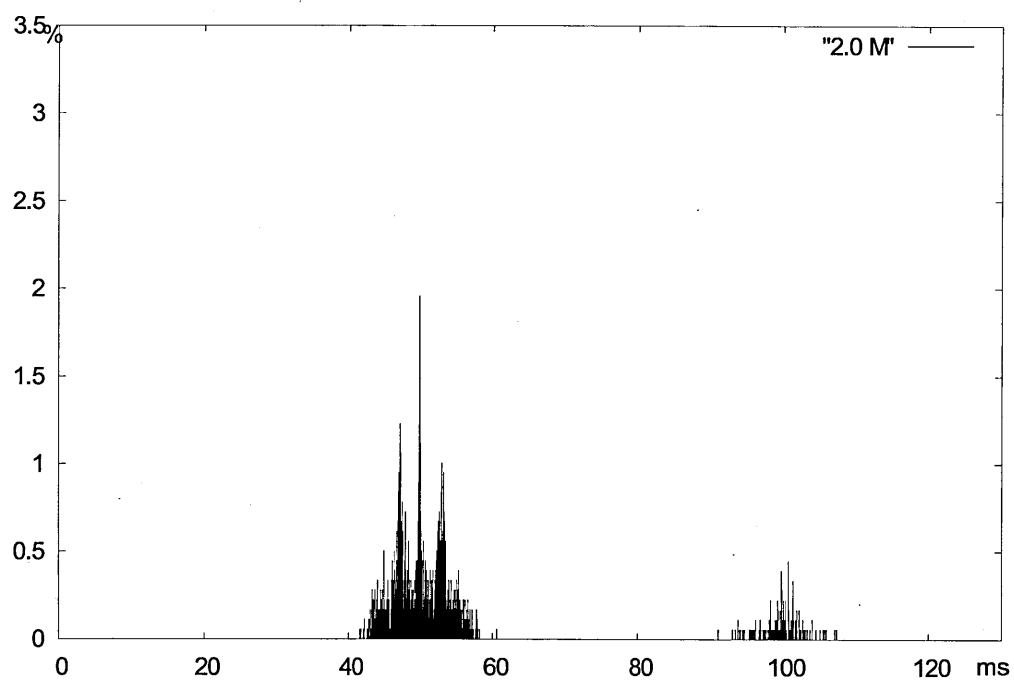


Figure 28 Server inter-arrival time with bandwidth 2 MB

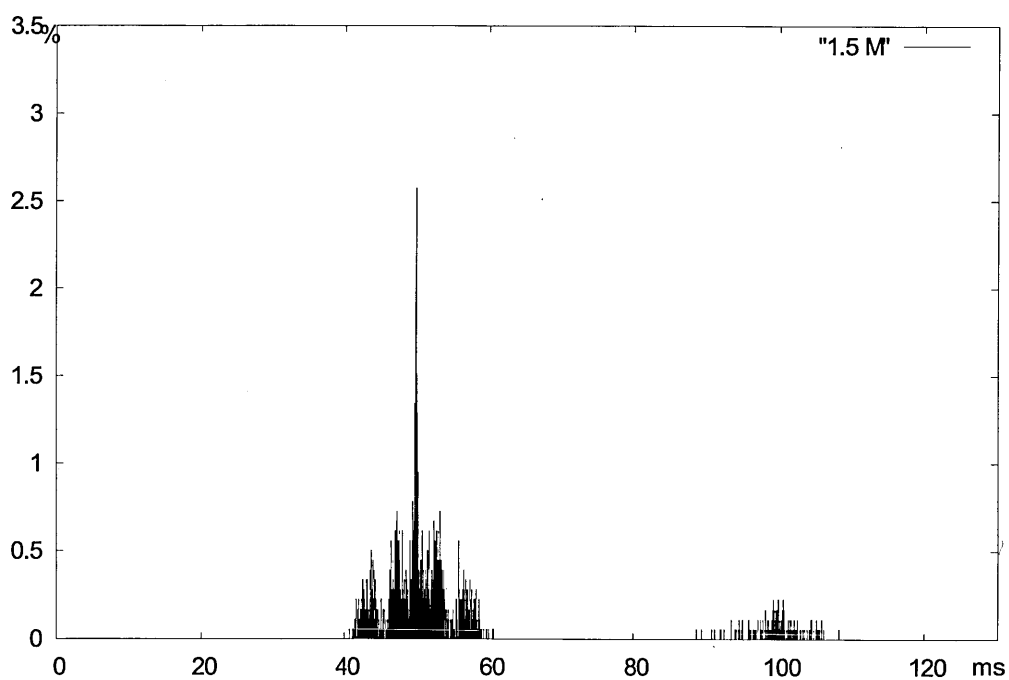


Figure 29 Server inter-arrival time with bandwidth 1.5 MB

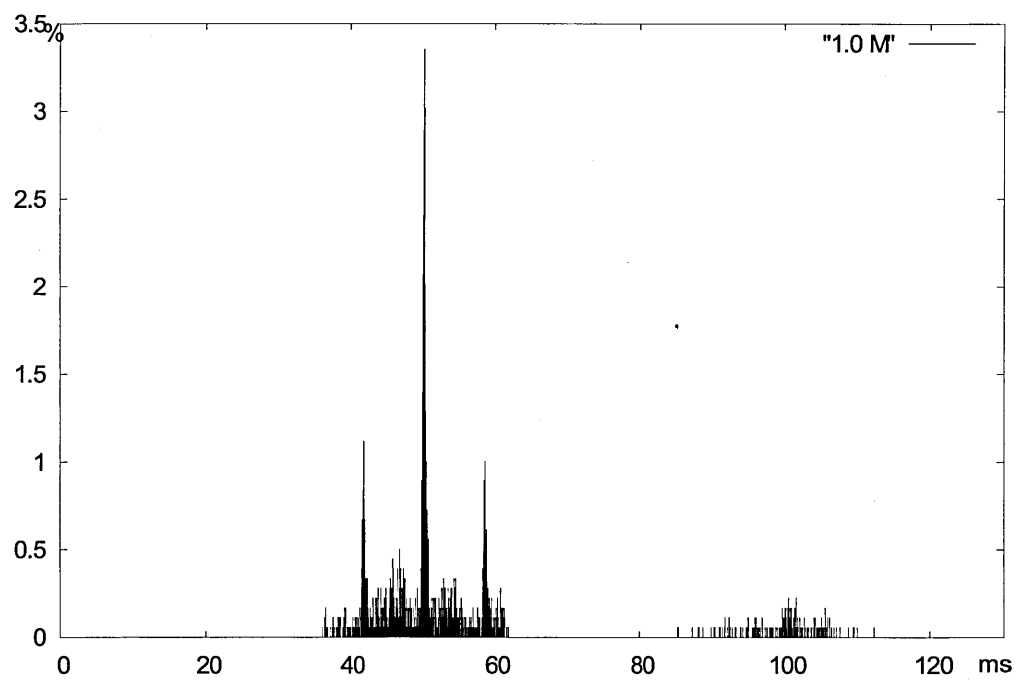


Figure 30 Server inter-arrival time with bandwidth 1 MB

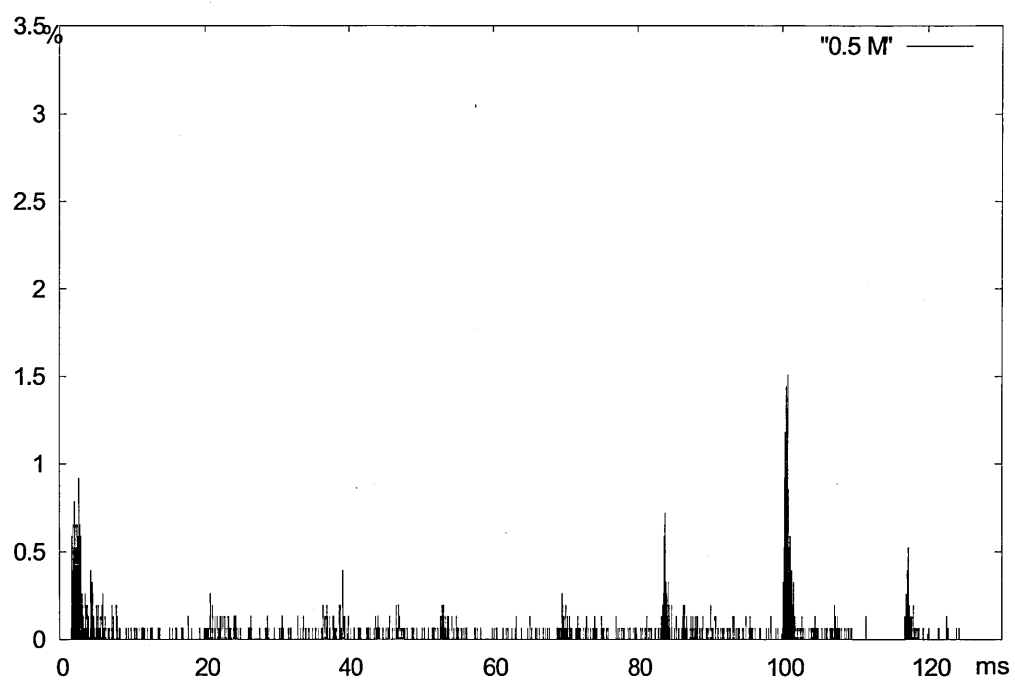


Figure 31 Server inter-arrival time with bandwidth 0.5 MB

When bandwidth was big (the bandwidth of 2.5 MB), game packets were re-transferred soon after they arrived any node in the network, the server inter-arrival time plot did not change the original dual-peak characteristic of the server inter-send time plot.

When bandwidth decreased, more and more game packets were buffered, the server inter-arrival time became more and more scattered around the two major peaks. When bandwidth was 1.5 MB, the two peaks fully disappeared (they were replaced by a single peak in the middle location of the two original peaks), dual-peak became an approximate bell. After this simulation result was compared to the realistic game traffic with other traffic, the simulation result had approximately similar bell-shape as the Internet experiment result. The problem about the disappearance of the server inter-packet time's dual-peak in the Internet experiment was solved. This simulator experiment built a bridge between the test-bed experiment and the Internet experiment.

Further decrease of bandwidth (the bandwidth of 0.5 MB) caused heavy network congestion, the result showed a highly scattered inter-arrival time. Some inter-arrival time dropped below the minima observed in both "clean" and realistic networks, some packets had inter-arrival time of nearly zero. Many game packets were release from buffering at little interval time, this can be attributed to the nature of the buffering and releasing from queuing of the game traffic together with TCP flows in the simulator. Because no similar empirical result of the extremely low inter-arrival time was observed in the realistic Internet experiment, it is reasonable to assume that the network congestion was not so heavy when the Internet experiment was performed.

3.3.2. EFFECTS FROM DIFFERENT TRAFFIC

The objective of this experiment was to explore effects on the game traffic from different types of other traffic. There were five different types of traffic (CBR, FTP, EXP, PAR and WEB) in the simulation. The same simulator experiment was repeated five times (sessions), with different type of the other traffic in every session. The game traffic was generated by replaying the realistic collected game traffic from the test-bed experiment, this guaranteed the same game traffic from the source nodes in

all five sessions. The behaviour of game server inter-arrival time and server-to-client delay with different types of other traffic was obtained and analyzed.

All parameters of the simulated network were same in the five simulation sessions: the bandwidth of the shared link was 200 KB, the queue length on both end nodes of the shared link was 64 (the selection of 64 was based on many tries because lower values, for example 32, caused more game packets to drop). The packet size of other traffic was 540 bytes or the nearest possible value. The streams of CBR, EXP, and PAR can be fully controlled by user program therefore their packet sizes were set to 540 bytes, the streams of WEB and FTP cannot be fully controlled by user program therefore the average packet size was calculated, the average packet sizes of WEB and FTP were 520 and 540 bytes, respectively. The duration of generating game traffic was long (100 seconds) to guarantee the number of game packets (1828 packets in every session of this simulation) is large for further statistical analysis. The experiment in every session was performed slightly longer than 100 seconds: packets of other traffic were already in queues along transfer path before game traffic started to transfer, and all temporarily buffered game packets had arrived at their sink node before simulation ended.

Because dropped packets changed the inter-arrival time behaviour, the rate of the other traffic was carefully adjusted to ensure that game packet-drop started to happen (or immediately before game packet-drop started to happen). Whereas the volume of CBR, EXP, and PAR can be explicitly adjusted in user program in ns2 simulator, the volume of FTP cannot be controlled in user program (it is automatically and dynamically controlled by the total volume of all traffic along the transfer path), and the volume of WEB cannot be fully controlled in user program. It was very difficult to achieve exactly the same volume of all these five types of traffic in this experiment, because of the difficulty in controlling volume of FTP or WEB (a slight change of the value of some network parameter sometimes caused big change of the total volume of FTP or WEB). Ensuring game packets started to drop in all five simulation sessions added difficulty to achieve the same volume of five types of other traffic. The original design of this experiment and its data analysis was finally changed, therefore the five types of other traffic were divided into two groups: EXP, PAR and WEB were in Group One, CBR and FTP were in Group Two. The game traffic behaviour was compared only within each group.

The parameters of this simulator experiment and part of results (number of dropped packets of game traffic) are shown in the next table. The maximum number of dropped packets of game traffic was 14 (in the case of CBR), it was small compared to the total number of game packets (1828). The traffic rate of CBR or FTP (Group One) was 208 KB, the traffic rate of EXP, PAR or WEB (Group Two) was between 140 and 144 KB. The “noise” in the top row of the next table represents “other traffic”.

	Type of noise	Num of noise pkts	Num of noise drop	Num of game drop	Noise packet size excl header			Total noise bytes excl h	Mean noise rate(k) excl h	Mean noise rate(k) incl. h
					Min	Max	Mean			
Group 1	CBR	4843	36	14	540	540	540	2615220	193	208
	FTP	4743	0	0	40	1040	540	2562720	189	208
Group 2	EXP	3289	0	0	540	540	540	1776060	131	141
	PAR	3343	36	11	540	540	540	1805220	133	144
	WEB	3305	38	14	40	1040	520	1721200	127	140

Table 11 Parameters of other traffic in simulation

3.3.2.1. SERVER INTER-ARRIVAL TIME

The plots of game server inter-send time had sharp dual-peak shape. The plots of inter-arrival time in the results of this simulation experiment somehow had the dual-peak shape although the peaks were not so sharp as in the server inter-send time plots. Some simple statistics (for example, average, variance) cannot apply directly because this dual-peak distribution, therefore Mann-Whitney test was used.

Mann-Whitney test result is related to sample-size, and Mann-Whitney test cannot apply directly to very large sample-size. For every type of traffic, five sets of samples (thirty samples in every set) of game server inter-arrival time were randomly selected for the Mann-Whitney test. Mann-Whitney test was performed with samples from impact of CBR against samples from impact of FTP (CBR-FTP) in Group One. Mann-Whitney test was performed with samples from impact of EXP against samples from impact of PAR (EXP-PAR), and also EXP-WEB, PAR-WEB in Group Two. The results of Mann-Whitney test are shown in the next table.

	Noises	Sampling 1	Sampling 2	Sampling 3	Sampling 4	Sampling 5
Group 1	CBR - FTP	0.052	0.114	0.003	0.007	0.297
Group 2	EXP - PAR	0.641	0.549	0.483	0.036	0.750
	EXP - WEB	0.865	0.501	0.145	0.011	0.947
	PAR - WEB	0.888	0.333	0.429	0.134	0.559

Table 12 Mann-Whitney test of server inter-arrival time with different traffic

As to the server inter-arrival time with the traffic of CBR and FTP in Group One, two of the five results of Mann-Whitney test were less than 0.05 (showing the impacts from CBR and FTP were different), however, the rest three of the five results were larger than 0.05. The results from Mann-Whitney were not able to support that the impacts on the server inter-arrival time were different. The similar situation was in the case of Group Two (EXP- PAR, EXP-WEB).

Although different traffic had slightly different effects on game traffic inter-arrival time behaviour, no sufficient empirical results supported that the server inter-arrival times with impacts from different types of traffic had different medians or distributions.

3.3.2.2. SERVER-TO-CLIENT DELAY

The end-to-end delay from server to client is server-to-client delay, and server-to-client delay was explored in this simulator experiment.

The server-to-client delay included the propagation delay. The propagation delay was the same to every packet and was 44 ms due to experiment design in Chapter Two. After the propagation delay was taken away from the server-to-client delay, the delay (in ms) by non-propagation factors in the five simulation sessions was obtained and its basic statistics are shown in the next table.

	Noise	Min	Max	Average
Group 1	CBR	43.602	1124.922	713.113
	FTP	1.624	690.212	293.084
Group 2	EXP	1.848	807.879	156.970
	PAR	1.624	1071.843	176.911
	WEB	1.624	1470.220	258.440

Table 13 Server-to-client delay with different traffic

Although all network parameters were same for the traffic of CBR and FTP, the average game packet server-to-client delay was different (713 and 293 ms, respectively) with impact from each traffic. The similar situation was between the traffic of EXP and WEB (and between PAR and WEB). Based on these empirical results, traffic streams with the same parameters but different inter-packet time characteristics had different impacts on game traffic end-to-end delay.

In the rows of “Group 2” in the above table, the average game packet delay with traffic of EXP and PAR is 156 and 176 ms, respectively. The difference of 156 and 176 is relatively small compared to the difference of average game delay of 258 by non-UDP traffic (WEB) in “Group 2”, and compared to the difference of average game delay (713 and 293 ms) between UDP (CBR) and non-UDP (FTP) traffic. Different types of UDP traffic (for example, EXP and PAR) caused similar extra delay of game packets. While exploring the impact on the throughput of TCP by UDP, Joyce (2000) found that the underlying UDP protocol behaved in much the same manner regardless of the application that was running it. The empirical results in the above table confirmed Joyce’s conclusion with impact on game traffic by UDP: different type of UDP traffic limited the transfer of game traffic in much the same way.

3.4. EVALUATION OF ANALYTICAL TRAFFIC MODEL

3.4.1. APPROXIMATE AND FINE TUNED MODELS

An approximate analytical traffic model was obtained based on the game traffic analysis: the server packet size had extreme distribution; the server inter-send time had deterministic distribution: half packets had the inter-send time of 47 ms, nearly half packets had the inter-send time of 53 ms, some packets had the inter-send time of 100 ms; the client packet size was distributed between 39 and 59 bytes; the client inter-send time approximately had the dual-distribution of the deterministic and the uniform: one-third of inter-send times were about 33 ms, two-thirds of inter-send times were approximately uniformly distributed between 31 and 50 ms.

However, game network traffic is variable and this model is approximate. There was setback in the results of data analysis when samples from pseudo traffic by this approximate analytical model were analysed with samples from the original empirical game traffic. The cause to the setback was found and there were mainly two problems: one was that the analytical distribution was not fine tuned, the other was that empirical game traffic in different game sessions (and different clients) was slightly different. Modification was made to improve the precision of the analytical traffic model.

3.4.1.1. FINE TUNING OF ANALYTICAL MODEL

The analytical distribution of the metric of inter-packet time needed fine tuning, an example is the analytical client inter-send time distribution. It approximately had the dual-distribution of the deterministic and the uniform. The result of evaluating the similarity between the inter-send time by this dual-distribution and the original empirical inter-send time was not ideal. This dual-distribution is not fine tuned although it reflects the primary characteristics of the client inter-send time. A few fine tuned distributions were tried to replace this analytical distribution. One effective fine tuning was slot-based as follows:

Because the client inter-send time was mainly distributed between 31 and 55 ms, the interval from 31 to 55 ms was divided into fixed-length sub-intervals (slots). If the length of every slot was set to 50 μ s, the slots were [31, 31.05], [31.05, 31.1], [31.1, 31.15], ..., [54.95, 55] ms. The percentage of inter-send time on every slot was calculated from the original empirical traffic, a list of the percentage of slots became the modified analytical distribution of the client inter-send time.

3.4.1.2. ANALYTICAL DISTRIBUTION PER-CLIENT

Another problem of analytical this model was that game traffic to or from different clients, or in different game sessions was slightly different. An example was the server packet size. The server packet size approximately had extreme distribution. Extreme distribution was determined by its location and scale parameters. In order to obtain a sole extreme distribution for empirical traffic in all sessions and to all clients, the average of each parameter of empirical data (55.6 and 16.2, respectively, see Table

6 of Section 3.1.5) was used. The samples randomly selected from every of the thirteen sets of original empirical data of the server packet size were analyzed with the random samples generated by this extreme distribution. Mann-Whitney test was performed with random samples from every of the thirteen sets of empirical data against the random samples by this extreme distribution. The test results are shown in the next table, the cells with test result less than 0.05 are shaded.

Session	1				2				3				
Client	1	2	3	4	1	2	3	4	1	2	3	4	5
Sampling 1	0.589	0.894	0.952	0.790	0.620	0.300	0.290	0.086	0.501	0.280	0.178	0.162	0.185
Sampling 2	0.678	0.277	0.344	0.217	0.455	0.037	0.038	0.004	0.773	0.870	0.711	0.717	0.662

Table 14 Mann-Whitney test of server packet size with sole parameter

The two populations have different medians if the result of Mann-Whitney test is less than 0.05. The test results in the table did not support that the pseudo and realistic packet size had same median, because there are three shaded cells out of a total of twenty-six cells. The setback in the test result was caused by that the empirical server packet size distribution depended on game session (see figures in Section 3.1.5). The extreme distribution with sole value of the parameters did not fit to the empirical packet size of all sessions. Therefore the analytical distribution of the server packet size was modified and one such distribution was obtained per client. There were thirteen analytical distributions of the server packet size because there were thirteen sets of original empirical traffic in the test-bed experiment. All thirteen analytical distributions were extreme distribution but with slightly different location parameter and scale parameter.

3.4.1.3. ANALYTICAL GAME TRAFFIC MODEL

The slot-based fine tuning improved the precision of the analytical game traffic model. The empirical traffic distributions shown in Figure 5 and 7 in Section 3.1 are already with the interval of 50 μ s. Therefore, the analytical client inter-send time distribution is already shown in Figure 5, the analytical server inter-send time distribution is already shown in Figure 7. The analytical client packet size distribution

is also slot-based because it is the percentage on different bytes, it is already shown in Figure 8.

As to the server packet size, unlike the rest three metrics, its analytical distribution can be expressed by a single consecutive function (extreme distribution), although only discrete value are valid when the distribution is used to generate the server packet size. As discussed above, the extreme distribution of the server packet size to different clients had slightly different location and scale parameters, therefore there were thirteen extreme distributions in the modified analytical model. One of the extreme distributions is shown in the next figure.

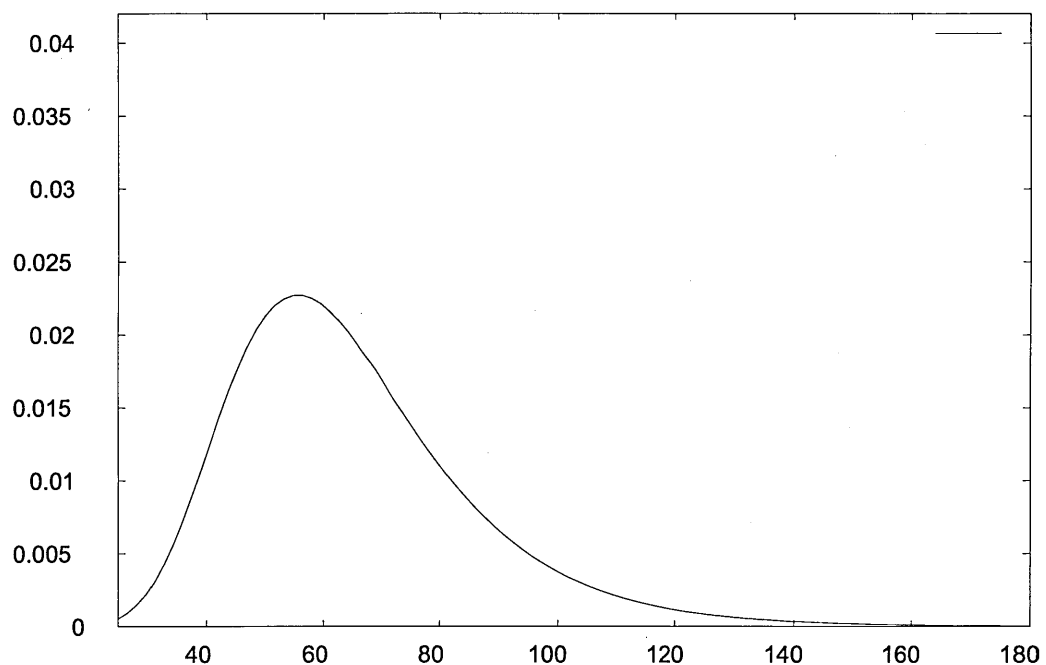


Figure 32 Analytical server packet size distribution

3.4.2. EVALUATION OF TRAFFIC METRICS

In order to evaluate the analytical game traffic model, metrics of pseudo game traffic by the analytical distribution were analysed not only for traffic generation, but also for the change through network transfer. Therefore not only inter-send time, but also inter-arrival time were analysed. When there is no or only a little packet drop during

transfer, the characteristics of packet size will not change. Therefore the packet size was only analysed for traffic generation.

In order to analyse the metric of inter-arrival time by the analytical distribution, a simulator experiment was performed. The simulation had the same network configuration as the rest simulations in this research, but with different selection of the other (background) traffic. Simulator provided the same network and the same other traffic for both pseudo and realistic game traffic to transfer, and same simulation was repeated twice with the pseudo and realistic game traffic, respectively. The pseudo game traffic consisted of randomly-selected (generated) samples by the analytical model, the realistic game traffic was generated by replaying the original empirical game traffic trace. Game traffic was collected in its network sink and its inter-arrival time was parsed.

As discussed before, ns2 simulator can generate five different types of traffic, they are CBR, FTP, EXP, PAR and WEB. It was found in this research that CBR usually does not have so heavy impact on the game traffic inter-packet time as EXP or PAR, whereas EXP or PAR usually do not have so heavy impact on the game traffic inter-packet time as FTP or WEB. Impact on game traffic from EXP or PAR is middle compared to those of other types of traffic, therefore either EXP or PAR can be selected. However, EXP is more well-known and widely used than PAR, therefore EXP was selected as the other traffic to investigate the inter-arrival time characteristics of pseudo game traffic in simulation. In order to simulate the competition for network resources between the game and the other traffic, the volume of EXP and the bandwidth of network were adjusted so that game traffic packets just started to drop.

After simulation was completed and the metric of inter-arrival time was parsed, samples of inter-arrival time were randomly selected. Mann-Whitney test was performed with samples of the inter-arrival time of the pseudo game traffic against those of the realistic game traffic.

In the last section, it is discussed that the sole value of parameter of extreme distribution needs modification. The case of the server packet size was just an example of the difference among sessions or clients. There was similar difference in the client packet size, in the client inter-send time and in the server inter-send time

after examining the empirical data. Therefore, the analytical distribution of every of these four metrics was obtained per client in data analysis.

Mann-Whitney test was performed in data analysis. Unless otherwise noted, ten sets of samples (thirty samples in every set) of both the analytical and the realistic traffic of every client were randomly selected and grouped into pairs (one analytical set versus one realistic set) for the Mann-Whitney test. The number of the total test results is 130 because they were thirteen original client traffic traces and ten tests were performed per client.

3.4.2.1. CLIENT PACKET SIZE

In order to evaluate the analytical distribution of the of the client packet size, Mann-Whitney test was performed with samples by the analytical client packet size distribution against samples from the client packet size empirical data. The test results are shown in the next table.

Session	Client	Sampling									
		1	2	3	4	5	6	7	8	9	10
1	1	0.4426	0.3608	0.7455	0.1544	1.0000	0.7036	0.3187	0.9940	0.2183	0.6104
	2	0.2264	0.2977	0.1731	0.9290	0.9820	0.4482	0.2503	0.2163	0.9229	0.8760
	3	0.9409	0.9467	0.4580	0.5423	0.6723	0.4833	0.1661	0.0408	0.7266	0.8588
	4	0.4804	0.7947	0.7775	0.6887	0.6190	0.5767	0.1519	0.8762	0.5874	0.5628
2	1	0.0806	0.4354	0.6757	0.7607	0.7827	0.4750	0.9940	0.2243	0.4180	0.7836
	2	0.1838	0.9231	0.0392	0.9823	0.7106	0.8762	0.1656	0.9350	0.6332	0.7894
	3	0.1505	0.9882	0.3659	0.7277	0.2412	0.6941	0.2386	0.1399	1.0000	0.7329
	4	0.5094	0.3276	0.2039	0.8067	0.7613	0.7271	0.2477	0.5000	0.9822	0.2759
3	1	0.3940	0.9281	0.4863	0.6494	0.9761	0.1427	0.5208	0.8637	0.3346	0.4379
	2	0.5821	0.6606	0.8755	0.4530	0.3270	0.4000	0.3933	0.7770	0.6050	0.5742
	3	0.4582	0.4668	0.2217	0.6820	0.1159	0.5271	0.1571	0.3194	0.7487	0.5826
	4	0.1518	0.8519	0.9525	0.3926	0.6668	0.5429	0.9108	0.0353	0.6182	0.1170
	5	0.0787	0.3210	0.5932	0.1817	0.8470	0.4344	0.5024	0.0997	0.0514	0.7893

Table15 Mann-Whitney test of pseudo client packet size

Almost all results in the above table are larger than 0.05, therefore the client packet size distribution and the empirical client packet size approximately had the same median.

3.4.2.2. CLIENT INTER-SEND TIME

As discussed above, the analytical distribution was approximate and the slot-based fine tuning was used. In order to achieve the best slot-based fine tuning, different slot

intervals were tried and they were 10, 20, 50, 500 and 5000 μ s, respectively. The number of non-compliant out of the 130 Mann-Whitney tests with different slot intervals are shown in the next table, the slot interval 50 has the least non-compliant cells of Mann-Whitney test.

slot interval (μ s)	10	20	50	200	500	5000
non-compliant	8	6	5	10	14	97

Table16 Number of non-compliant of Mann-Whitney test of client inter-send time

The details of the Mann-Whitney test results with slot interval 50 are shown in Table 17.

Session	Client	Sampling									
		1	2	3	4	5	6	7	8	9	10
1	1	0.4642	0.9000	0.2706	0.0656	0.0149	0.3402	0.6307	0.6843	0.5592	0.8073
	2	0.6520	0.7506	0.1624	0.2611	0.4119	0.5298	0.0992	0.6520	0.5011	0.3711
	3	0.9587	0.2904	0.3328	0.4290	0.8302	0.1297	0.7283	0.9470	0.3953	0.6204
	4	0.7224	0.7061	0.9469	0.9233	0.9941	0.9350	0.2335	0.7844	0.4732	0.2703
2	1	0.4547	0.9587	0.0848	0.3038	0.0107	0.7728	0.1903	0.3790	0.8592	0.7393
	2	0.3554	0.2397	0.4918	0.3110	0.3789	0.6734	0.9469	0.4643	0.2062	0.5298
	3	0.5493	0.5996	0.9705	0.6843	0.2061	0.2517	0.6099	0.3871	0.0615	0.4376
	4	0.4368	0.6520	0.5683	0.9763	0.2325	0.6727	0.3175	0.3554	0.0066	0.8185
3	1	0.2953	0.8072	0.8529	0.4193	0.8185	0.4352	0.9234	0.6838	0.9528	0.7505
	2	0.8072	0.1295	0.3669	0.7956	0.2829	0.5994	0.4638	0.7731	0.8766	0.8302
	3	0.6307	0.8302	0.1759	0.3510	0.0790	0.9234	0.0989	0.1453	0.3710	0.8418
	4	0.2097	0.9116	0.2375	0.7050	0.4277	0.2374	0.7277	0.2757	0.1621	0.5294
	5	0.6401	0.5996	0.0297	0.8183	0.0480	0.5281	0.6623	0.5488	0.0190	0.8533

Table17 Mann-Whitney test of pseudo client inter-send time

There are 130 results of Mann-Whitney test in the above table. Only 5 results are less than 0.05 (non-compliant), the rest 125 results are larger than 0.05, therefore the client inter-send time distribution and the empirical client inter-send time approximately had the same median.

3.4.2.3. CLIENT INTER-ARRIVAL TIME

In order to evaluate the client inter-arrival time in the pseudo game traffic by the analytical model, Mann-Whitney test was performed with random samples of the client inter-arrival time from the pseudo game traffic against random samples of the client inter-arrival time empirical data. The test results are shown in the next table.

Session	Client	Sampling									
		1	2	3	4	5	6	7	8	9	10
1	1	0.3076	0.6952	0.5201	0.1785	0.9234	0.6414	0.8130	0.7788	0.8650	0.3147
	2	0.5997	0.5493	0.5997	0.2675	0.3328	0.3871	0.0501	0.8534	0.4119	0.7506
	3	0.5692	0.4204	0.0163	0.3555	0.5894	0.0635	0.6735	0.3555	0.7845	0.4376
	4	0.5894	0.8418	0.2905	0.6309	0.4825	0.2939	0.4733	0.3183	0.8418	0.3790
2	1	0.8883	0.3042	0.9000	0.9705	0.3478	0.4825	0.2170	0.0919	0.3953	0.2170
	2	0.8883	0.1413	0.4688	0.4825	0.4918	0.2905	0.3403	0.4733	0.2062	0.9587
	3	0.0635	0.1120	0.0701	0.8534	0.3632	0.1958	0.9352	0.8073	0.1055	0.4918
	4	0.4376	0.8592	0.9411	0.1494	0.8592	0.7618	0.8418	0.5298	0.9823	0.0002
3	1	0.4077	0.1278	0.0351	0.8883	0.6789	0.3255	0.4598	0.3994	0.9352	0.5692
	2	0.0760	0.3255	0.7172	0.0303	0.4464	0.7394	0.8534	0.0850	0.1335	0.3329
	3	0.6309	0.5742	0.4290	0.4376	0.2062	0.6414	0.8418	0.7283	0.1171	0.2838
	4	0.4779	0.9352	0.9176	0.7731	0.7958	0.2707	0.1154	0.0378	0.9823	0.0635
	5	0.4733	0.1809	0.3183	0.2398	0.0117	0.5692	0.3593	0.5249	0.8360	0.8941

Table18 Mann-Whitney test of pseudo client inter-arrival time

There are 130 results of Mann-Whitney test in the above table. Only 5 results are less than 0.05 (non-compliant), the rest 125 results are larger than 0.05, therefore the client inter-arrival time of pseudo traffic and the empirical client inter-arrival time in the simulation approximately had the same median.

3.4.2.4. SERVER PACKET SIZE

The analytical distribution of the server packet size was extreme distribution. After the setback of the sole value of the extreme distribution, the scale and location parameters of extreme distribution were obtained per client. Mann-Whitney test was performed with random samples generated by this per-client distribution against random samples from the empirical server packet to the same client. Mann-Whitney test results are shown in the next table.

Session	Client	Sampling									
		1	2	3	4	5	6	7	8	9	10
1	1	0.3061	0.3781	0.0959	0.3355	0.7385	0.9528	0.5887	0.1623	0.6305	0.2897
	2	0.3424	0.7501	0.0770	0.7727	0.1461	0.6252	0.8242	0.2812	0.3435	0.3625
	3	0.5333	0.8531	0.0283	1.0000	0.0053	0.5688	0.7056	0.1134	0.3107	0.1114
	4	0.2382	0.7955	0.0113	0.8357	0.2922	0.1951	0.5241	0.2257	1.0000	0.2079
2	1	0.4187	0.2929	0.0960	0.6835	0.0839	0.5488	0.5001	0.0276	0.4771	0.3586
	2	0.2822	0.4324	0.3230	0.5734	0.0681	0.6894	0.7057	0.3765	0.5440	0.5340
	3	0.6037	0.5837	0.1715	0.7164	0.0966	0.6840	0.6999	0.0778	0.8415	0.6677
	4	0.4493	0.6093	0.1204	0.7222	0.1091	0.8765	0.3174	0.0506	0.8474	0.1254
3	1	0.3349	0.8939	0.0960	0.8881	0.1608	0.4818	0.5193	0.1035	0.7841	0.9351
	2	0.7162	0.7897	0.1958	0.3281	0.0566	0.6785	0.6041	0.0417	0.9764	0.2303
	3	0.3538	0.2860	0.2011	0.7897	0.0309	0.7955	0.5338	0.2555	0.7955	0.6408
	4	0.3462	0.4908	0.1786	0.8010	0.0966	0.3472	0.6197	0.1513	0.3587	0.9646
	5	0.3538	0.0778	0.0294	0.8705	0.0851	0.9469	0.7838	0.0697	0.8474	0.1385

Table 19 Mann-Whitney test of pseudo server packet size by extreme distribution

There are 130 results of Mann-Whitney test in the above table, seven results are less than 0.05 (non-compliant). In order to find a more fitting analytical distribution, the slot-based fine tuning was used and per-client analytical distribution was obtained based on the empirical data of packet size in different lengths (bytes). Mann-Whitney test was performed in the same way as the above extreme distribution and the results are shown in the next table.

Session	Client	Sampling									
		1	2	3	4	5	6	7	8	9	10
1	1	0.5104	0.0760	0.6099	0.3074	0.2338	0.6307	0.1333	0.7617	0.8071	0.5591
	2	0.5295	0.5691	0.3364	0.0129	0.8129	0.1277	0.1276	0.6203	0.7673	0.2518
	3	0.4962	0.9175	0.7561	0.0351	0.5056	0.3040	0.5057	0.7171	0.8128	0.5590
	4	0.8591	0.1153	0.9469	0.5640	0.2427	0.5441	0.4917	0.9646	0.4915	0.5393
2	1	0.8015	0.5741	0.1392	0.5996	0.4686	0.0723	0.5945	0.7226	0.5541	0.6788
	2	0.2224	0.8883	0.5842	0.8187	0.8187	0.7449	0.0747	0.2487	0.6046	0.8244
	3	0.5443	0.4732	0.7787	0.3217	1.0000	0.3592	0.5248	0.0933	0.8072	0.3477
	4	0.2672	0.6255	0.6046	0.5691	0.9000	0.6361	0.2280	0.6680	0.6466	0.7281
3	1	0.8999	0.9469	0.5993	0.7505	0.7003	0.7729	0.0770	0.1713	0.8301	0.4731
	2	0.1351	0.6570	0.4822	0.3869	0.5591	0.2970	0.8072	0.7225	0.6198	0.9410
	3	0.2140	0.1599	0.5344	0.7730	0.2900	0.6044	0.4417	0.6733	0.0674	0.8475
	4	0.5842	0.8244	0.5994	0.0784	0.7561	0.1955	0.2610	0.8648	0.2424	0.1905
	5	0.9292	0.3706	0.9116	0.0404	0.2970	0.9704	0.5056	0.1904	0.2731	0.6413

Table 20 Mann-Whitney test of pseudo slot-based server packet size

There are 130 results of Mann-Whitney test in the above table. Only 3 results are less than 0.05 (non-compliant), the rest 127 results are larger than 0.05, therefore the server packet size distribution and the empirical server packet size approximately had the same median.

Based on the result of Mann-Whitney test between the analytical and empirical server packet size, the server packet size by the extreme distribution or by the slot-based analytical distribution approximately had the same median as the empirical server packet size, although the slot-based analytical distribution had better fitting to the empirical data than the extreme distribution.

3.4.2.5. SERVER INTER-SEND TIME

In order to evaluate the analytical distribution of the of the server inter-send time, Mann-Whitney test was performed with samples by the analytical server inter-send time distribution against samples from the server inter-send time empirical data. The test results are shown in the next table.

Session	Client	Sampling									
		1	2	3	4	5	6	7	8	9	10
1	1	0.3110	0.6843	0.3327	0.8766	0.0484	0.8360	0.5692	0.8766	0.5009	0.2115
	2	0.8534	0.4375	0.1557	0.8302	0.6308	0.1373	0.2115	0.5395	0.0594	0.2282
	3	0.5894	0.8534	0.1086	0.7117	0.8766	0.6843	0.7338	0.1433	0.0518	0.5692
	4	0.1809	0.6413	0.0421	0.0271	0.5492	0.2518	0.8650	0.2837	0.5297	0.1579
2	1	0.2739	0.3830	0.8072	0.3041	0.5346	0.2062	0.6627	0.0977	0.3111	0.6735
	2	0.8418	0.3183	0.6680	0.5591	0.3911	0.5894	0.4376	0.1259	0.1760	0.9352
	3	0.9823	0.3183	0.0144	0.9705	0.8418	0.5011	0.7061	0.0905	0.1714	0.3994
	4	0.5894	0.3711	0.4732	0.7282	0.6308	0.0614	0.6573	0.7062	0.4161	0.6952
3	1	0.3182	0.6843	0.8187	0.8073	0.0904	0.5493	0.2518	0.5058	0.1907	0.6100
	2	0.9705	0.1296	0.4642	0.0635	0.5792	0.7618	0.5692	0.3291	0.7845	0.9705
	3	0.2062	0.3631	0.8883	0.1882	0.8883	0.5395	0.7844	0.4733	0.4035	0.4290
	4	0.0797	0.4824	0.1857	0.0215	0.9352	0.3478	0.5011	0.2707	0.1023	0.6204
	5	0.2518	0.4376	0.4203	0.2549	0.3328	0.4918	0.8418	0.7731	0.5395	0.3042

Table 21 Mann-Whitney test of pseudo server inter-send time

There are 130 results of Mann-Whitney test in the above table. Only 5 results are less than 0.05 (non-compliant), the rest 125 results are larger than 0.05, therefore the server inter-send time distribution and the empirical server inter-send time approximately had the same median.

3.4.2.6. SERVER INTER-ARRIVAL TIME

In order to evaluate the client inter-arrival time of the pseudo game traffic by the analytical model in the simulator experiment, the server traffic to one client was randomly selected. Five sets of samples (thirty samples in every set) of the server inter-arrival time of the pseudo game traffic and of the empirical data were randomly selected and grouped into pairs (one pseudo set versus one realistic set). Mann-Whitney test was performed with each pair set of samples. The test results are shown in the next table.

	Sampling 1	Sampling 2	Sampling 3	Sampling 4	Sampling 5
Test	0.529	0.297	0.600	0.378	0.756

Table 22 Mann-Whitney test of pseudo server inter-arrival time

The data in all cells in the above table are larger than 0.05, therefore the game server inter-arrival time by the pseudo and realistic traffic with other traffic in simulation had the same median.

CHAPTER 4

DISCUSSIONS, CONCLUSIONS AND FUTURE WORK

4.1. DISCUSSIONS

While most previous published game traffic research concerned only the “interfered” traffic, the “clean” and “interfered” traffic was analyzed together in this research.

While most previous published game traffic research concerned either modeling traffic or simulating traffic, the modeling and simulating were combined in this research, and a theoretical assumption about the discrepancy between “clean” and “interfered” traffic was confirmed by the simulator experiment.

4.1.1. UNDERSTANDING SERVER INTER-SEND TIME

In the test-bed experiment, most server packets were sent at the interval of either 47 or 53 ms, some packets were sent at the interval of around 100 ms. The statistical results are shown in Table 4 in Chapter Three. The data in Column “Peak 3” of that table are irregular among the thirteen clients (rows) although the data in Column “Peak 1” and “Peak 2” are almost the same in all the thirteen rows. The cause to this behaviour of the server inter-send time was obtained after the original traffic traces were carefully examined.

A communication cycle between the Quake server and its clients is as follows:

A client communication cycle consists of reading a server packet, processing it, rendering players’ situations on the screen, sampling input devices, then transmitting an update packet to the server.

A server communication cycle consists of collecting data from every client, controlling game progress, and sending update packets to all (or most) clients one by one.

The packet sniffer Ethereal provides an interactive interface to examine every packet in a collected traffic trace. The original game traffic packet records were examined in the program Ethereal. The regular sending interval of any two consecutive packets from server to one client was either 47 or 53 ms. The server sent updating packets to the quickest player (for example, Client 2 in Session 1 and Session 2 in Table 4) in every communication cycle, therefore no server inter-send time to this quickest client was 100ms. However, all other clients were slower than the quickest client, the server did not send the updating packet to some slow clients in some communication cycles, therefore some inter-send time to the every of the other clients (except the quickest client) was approximately 100ms ($47 + 53 = 100$ ms).

4.1.2. UNDERSTANDING BASIC STATISTICS

The minimum, maximum and average packet size and inter-send time are shown in Table 10 in Chapter Three.

Server had bigger packets than client, because the average server packet size was 73.3 bytes and the average client packet size was 48.2 bytes. Server should tell a client more information (for example, all other clients' locations) in a single packet than what a client should tell server. The range between the minimum and maximum packet size of server was much larger than that of client (the packet sizes from client to server were less variable than that from server to client), the client packets were simpler than the server packets.

Server had longer inter-send time than clients, because the average server inter-send time (to one client) was 52.9 ms and the average client inter-send time was 37.1 ms. Server processed packets from all clients, controlled the game process, and sent different packets to different clients, therefore it likely took longer time before it sent the next packet to the same client. Game server had higher requirement of hardware than client.

Server did not stop sending packets to any client even though some client stopped sending packet to server in relatively short interval, because the maximum server inter-send time (199 ms) to one client was much smaller than the maximum client inter-send time (735 ms).

4.1.3. INTER-PACKET TIME AND QUEUING DELAY

Other traffic affected some game traffic metric behaviour heavily. A good example of the effects from other traffic was the server inter-packet time:

The server inter-send time plot had dual-peak (two major peaks) distribution. However, the dual-peak disappeared in the server inter-arrival time plot. Instead, the server inter-arrival time plot was an approximately bell-shaped plot in the corresponding location of the dual-peak. The deterministic (definite) inter-send time changed into a clustered (bell-shape) inter-arrival time. This change was not only observed in the realistic experiment in this research, but also generated (and observed) by in the simulator experiment. Therefore this change was typical and need further discussion.

This change was caused by non-identical (different) network transfer delay to every packet. In order to understand different network transfer delay, the composition of Internet delay is listed in Appendix C. Internet delay of game traffic between game server and client is end-to-end delay, it is the time required for a game packet to traverse the network from one game host computer (server or client) to the other. Understanding the end-to-end delay helps to locate the major cause to the above change.

Among components of Internet delay, the propagation delay is same to all packets therefore it will not change inter-packet time. The difference of transmission delay among different game packets (most packets are small) is negligible in high-speed network. Theoretically, the main cause to the change of inter-packet time is queuing delay.

The height of any peak in the inter-send time plot was decreased and the peak was changed into more scattered plot in the inter-arrival time. This can be explained by varying queuing delay. If packet arrival-sequence is the same as packet send-sequence, the varying of queuing delay to a single packet changes two values (one inter-time with the packet ahead and one inter-time with the packet behind) of inter-arrival time. If the varying of queuing delay to a single packet changes packet arrival-sequence, more than two values of inter-arrival time are therefore changed. The number of changed inter-arrival time is more than the number of changed queuing

delay. Although the number of changed values of inter-arrival time cannot be so high as doubling the number of packets with varied queuing delay if every queuing delay of a consecutive batch of packets varies, at least a batch of values of inter-arrival time do change.

Internet traffic can be very bursty and packets arrive in clumps. Assuming game traffic competes with bursty traffic on the Internet for transfer, the queuing delay increases because bursty traffic results in high bandwidth demands. The increase of queuing delay likely happens on a batch of game packets because game traffic is sent in high frequency (about 50 ms), therefore many values of inter-arrival time change. That is why the inter-arrival time distribution easily differed from the inter-send time distribution, no matter client was very near or far away from server in the Internet experiment, or no matter there was light or heavy competition for network resources between the game and the other traffic in the simulator experiment.

In case that the inter-arrival time between the first packet and the last packet remains the same from the inter-send time, the accumulation of the percentage of newly scattered plot (by varied queuing delay) in the left side of the peak equals to the accumulation of the percentage of newly scattered plot in the right side of the peak. Even though the inter-arrival time between the first packet and the last packet changes from the inter-send time, the accumulation of the percentage of newly scattered plot in the left side of the peak approximately equals to the accumulation of the percentage of newly scattered plot in the right side of the peak, as long as the number of total packets is high compared to this inter-arrival time (between the first packet and the last packet) change. That is why the all peaks of inter-send time plot changed into the bell-shape of inter-arrival time plot.

The simulator experiment offered solid support to that the queuing delay was the primary cause to the inter-packet time change. Ns2 simulator provides an animation tool NAM, the movement (for example, the number of packets being buffered in queue) of network packets in the simulation can be visually observed in the graphical interface of NAM. With the help of ns2 result interpretation and the display on NAM, the next progression was obtained:

When there was no other traffic in simulation, the game packet inter-arrival time was the same as the inter-send time. As the volume of other traffic gradually increased or

network bandwidth decreased, the inter-arrival time started to change and became different from the inter-send time. When lots of packets were buffered in the queue, the inter-arrival time behaviour changed noticeably from the original inter-send time behaviour.

The updating game packets should be exchanged periodically between client and server through network transfer. The game performance is likely impacted if packets arrive in a game computer in clumps in one period but no packet arrives in the computer in another period. Therefore the behaviour of inter-arrival time is important, and the role of queuing delay is critical.

4.1.4. CAUSES TO THE DISCREPANCIES

There were discrepancies of game traffic metrics in the previous published research results. The empirical results of the test-bed and Internet experiments suggested the discrepancies were caused by different networks or different traffic collection locations, and this situation was confirmed by the simulator experiment.

Almost all previous published empirical results of the game Quake traffic were obtained in the Internet experiments. The network configuration of the Internet is geographically related, and effect game traffic from other traffic is dynamic. Based on the empirical results in this research, the impact from other traffic (in different networks) was one primary cause to the discrepancies in previous publish results. However, some other possible causes likely contributed the discrepancies:

Different releases of the same game likely have different traffic behaviour. The empirical results in this research (Quake III) were less different from Lang et al.'s (2004) and Pavlicic's (2003) work on Quake III than from Borella's (2000) (on Quake I and Quake II) and Jorce's (2000) (on QuakeWorld) work. Therefore different releases of the same game likely have different traffic behaviour.

Player actions and computer hardware affect traffic behaviour. The empirical data of the client packet size or inter-arrival time from any single client in the test-bed experiment were composed of about 15000 effective packets, the data of the server packet size or inter-arrival time to any single client were composed of about 10000 packets because server had longer inter-send time (the least number of server's

packets was 9868). However, such number of packets per client did not offer the same metric plot nor the same metric average (for example, packet size, inter-packet time, packet rate, bit rate, etc) among different clients. These differences among different clients were not big. These minor differences were likely caused by factors in the player side (for example, host computer hardware, player actions), see next explanation.

In order to obtain game traffic characteristics in exceptional case, a very slow client host computer was connected to the test-bed network before the test-bed experiment was performed. A short pre-experiment was performed to prepare for the test-bed game experiment. A slow player played the game in this slow client host computer, while other client host computers were played by ordinary players. The client inter-send time of this slow client was different and irregular compared to that of the rest host clients, what is more, the server inter-send time to this slow client was also different and irregular compared to the server inter-send time to the rest host clients. The game server sent much less packets to very slow client (server did not send packet to that extremely-slow client in lots of communication cycles). Therefore game traffic was player action related. According to Borella (2000), some metric (client inter-arrival time) distribution was highly related with the CPU speed of the host computer. Therefore, game traffic is affected by factors in the player side. This was likely another possible cause to the discrepancies.

The game traffic behaviour was affected by map selection. It was already found in previous research and confirmed by the server packet size empirical data in the test-bed experiment: the three sessions of the test-bed experiment were played by the same players on the same network, the only difference among sessions was server map selection. The server packets size plots differed among different sessions (the differences among different players were minor compared to those among different sessions). Therefore the game traffic behaviour was game server-setting related.

4.2. CONCLUSIONS AND FUTURE DIRECTIONS

4.2.1. CONCLUSIONS

There are three objectives in Chapter one, the conclusions are organized by these three objectives.

4.2.1.1. GAME TRAFFIC CHARACTERISTICS

OBJECTIVE ONE in Chapter One:

The starting point for this research was to explore traffic characteristics (using the metrics of packet size and inter-packet time) generated by a FPS game. The first objective for this research was to explore:

- The “clean” game traffic and the relation of traffic caused by different server settings.*
- The “interfered” game traffic and the relation of traffic caused by different “interfered” network transfer paths.*
- The relation between the “clean” and “interfered” game traffic.*
- The relation of game traffic in different empirical observation locations, and the relation between packet inter-arrival and inter-send times.*

The FPS game Quake III Arena was selected and its traffic was analysed. The analytical distributions of the two fundamental metrics, packet size and inter-packet time, were successfully obtained. The empirical data of the game traffic were highly variable, therefore the obtained analytical distributions were approximate. The analytical distributions are as follows (see Chapter Three for details):

The client inter-send time approximately had the dual-distribution of the deterministic and the uniform: one-third of inter-send times were about 33ms, two-thirds of inter-send times were approximately uniformly distributed between 31 and 50 ms (except 33 ms).

The server inter-send time approximately had deterministic distribution: half packets had inter-send time of 47ms, nearly half packets had inter-send time of 53ms, some (about 6%) packets had inter-send time of 100ms.

The client packet size were approximately distributed in the interval between 39 and 59 bytes, 43-byte formed a minor peak compared to all other packet sizes because it had about 16% of total packets.

The server packet size approximately had extreme distribution.

Packet size and inter-send time were dependent on the host computer hardware; both client inter-send time and server inter-send time for a client were dependent on the player's action on that client; the server packet size was dependent on the game's map selection.

The average packet sizes of client and server were 73 and 48 bytes, respectively. Therefore the protocol overhead of game packet was high. Because the current design of routers tends to improve more the volume of throughput instead of reducing the delay of small packets, the tendency of game popularity forms a new request in router's design.

Based on the inter-send time distribution, almost all packets are expected to be dispatched from every client between every 31 and 50 ms, there is therefore at least one such packet at every interval of 50 ms. Compared to client, packets from the server can be predicted more precisely: to any single client, packets are expected to be dispatched from server at every 47 or 53 ms, with some single packet missing the dispatching chance occasionally. In order for server to render the process of gaming properly, server's data are supposed to be updated by any client in no more than every 50 ms. For the proper performance of client, client's data are supposed to be updated by server in no more than every 100 ms. Although game server allows clients with various server-client delay to join gaming in Quake III Arena (the player successfully joined servers with different server-client delay and played the game in the Internet experiment, the shortest server-client delay was 12 ms and the longest server-client delay was 265 ms), because of the above designed frequency of game updating packets, if the difference of the server-client delay between any two clients surpasses certain threshold, say 100 ms, the player with longer delay is theoretically disadvantaged in gaming.

Based on the minimum and maximum inter-send time of the test-bed experiment, server did not stop sending packets to any client even though some client stopped sending packet to server in relatively short interval, because the maximum server inter-send time to one client was much smaller than the maximum client inter-send time.

The server inter-arrival time from farther away servers usually had higher percentage around 100 ms. A server sent less packets to farther away clients, more packets were dropped during network transfer from a farther away server.

The characteristics of the server inter-packet time of non-team and team game types are different. The team types have longer inter-arrival time than non-team type.

Although the game traffic behaviour was the same in different observation locations in the “clean” network, it was different in the arrival end between the “clean” and “interfered” network, or among different “interfered” networks. The game traffic behaviour was dependent on observation location in “interfered” network, or network environment. The characteristics of inter-arrival and inter-send time can be quite different.

It is a shortcoming that some previous research (details see Chapter One) offered a relatively-precise analytical distribution of inter-packet time without considering the possible change of inter-packet time between the sending and arrival points, or did not consider empirical traffic collection location.

4.2.1.2. EFFECTS ON GAME FROM OTHER TRAFFIC

OBJECTIVE TWO in Chapter One:

The second objective of this research was to explore the role of other traffic. It was to build an experiment in order to show the role of other traffic in the progression from the “clean” to the “interfered” game traffic behaviour. It was also to explore the impacts on game traffic metrics from different types of traffic.

There was noticeable difference of the inter-arrival time between the test-bed and the Internet experiments. A simulator experiment built a bridge between the “clean” (dual-peak) and the typical “interfered” (bell-shape) inter-arrival time and showed the

progression of inter-packet time distribution from the “clean” to the realistic traffic. This observation offered the answer to the disappearance of server inter-packet time’s dual-peak in the realistic experiment. Other traffic was the primary cause of the difference of inter-packet time between the test-bed and the Internet experiments, and was likely one of the primary causes to the discrepancy of inter-packet time behaviour in the previous published results.

It is the varying queuing delay among all packets that decreased the height of any peak in the inter-send time plot and changed the peak into more scattered bell-shape around the peak in the inter-send time plot.

When there is sufficient bandwidth for game traffic to traverse, the game server inter-packet time has dual-peak. As bandwidth decreases or volume of other traffic increases, the game traffic and other traffic compete for network resources, the queuing delay of game packet increases. If the increased queuing delay is the same for every packet, the inter-packet time will not change. Many sessions of the Internet experiment and of the simulator experiment were performed, the dual-peak in the inter-arrival time definitely changed in any case, as long as there was competition from other traffic. Based on the empirical data of this research, it is not possible for queuing delay to every game packet to be identical in the presence of competition from other traffic.

There are two peaks with about 6 milliseconds apart each other in the plot of the server inter-send time. Assume that the difference of queuing delay among packets is a few milliseconds. As the height of both peaks decrease and plot becomes scattered around each peak in the presence of varying queuing delay, the interval between the two peaks is partly or totally overlapped by scattered plots from both peaks, whereas the areas on both sides of this interval ([47, 53] ms) are not overlapped or not so heavily overlapped because the plot in either side is at least 6 ms away from one peak. The inter-arrival time inside this interval therefore has higher percentage than that outside this interval in the histogram plot. The dual-peak approximately turns into a single bell-shape. There were many sessions in the Internet experiment. The difference of queuing delay among most game packets was in the order of magnitude of milliseconds in almost all sessions the Internet experiment, based on all figures in Section 3.2.2.2. The conclusion of the milliseconds order of magnitude of varying

queuing delay among game packets can be extended to other applications with similar Internet end-to-end delay because the game traffic did not receive any special service in the Internet experiment.

A simulator experiment showed that other traffic streams with same parameters except inter-packet time characteristic had different effects on game traffic end-to-end delay. These different effects represent that some traffic has heavy impact on game traffic metric behaviour, whereas some has light impact. That is, although the data rate or throughput of the other traffic is the same, the effects on game traffic end-to-end delay from such other traffic can be different. The simulation results in Section 3.3.2.2 showed this difference: the on-off traffic (EXP or PAR) added smallest average end-to-end delay to game traffic, then came bursty traffic (FTP), and the constant rate traffic (CBR) added largest additional average end-to-end delay to game traffic. It is because game packets are sent in high frequency therefore any off-period of the other traffic provided game traffic a period of “un-interfered” network link. As discussed above, queuing delay is the primary contribution to the game traffic end-to-end delay, the additional game traffic delay during this off-period is very short, therefore the average game traffic end-to-end delay is smaller than that in the presence of other types of traffic. Based on this empirical result, in some extreme case when different paths are dominated by different types of other traffic, selecting network path smartly due to the type of dominating traffic can likely help to decrease game traffic delay.

4.2.1.3. APPLICABILITY OF ANALYTICAL MODEL

OBJECTIVE THREE in Chapter One:

The third objective of this research was to evaluate the usability of an analytical traffic model, and the possibility of replacing realistic game traffic with the pseudo traffic generated by the analytical model.

The pseudo game traffic was generated by the analytical traffic model obtained in the test-bed experiment. The metrics of inter-packet time and packet size combined are able to characterize realistic traffic or create pseudo traffic. By selecting slightly fine tuned model of each metric per client, the pseudo traffic behaved approximately the same way as the realistic traffic in the simulator experiment, based on the analysis of

the pseudo game traffic before and after transfer in simulator network with other traffic. It is therefore possible for game developers to employ pseudo game traffic to replace realistic game players in some experiments to decrease development period or cost. Similarly, game server providers would benefit in using the results when configuring their servers.

4.2.2. FUTURE WORK AND POSSIBLE DIRECTIONS

4.2.2.1. MODELLING METHOD

The empirical data are usually large and highly variable in network traffic experiments, it is difficult to determine the analytical distribution of network traffic metric. Because of this special difficulty, deleting some unexpected empirical data (packets) was done by some previous researchers (for example, Paxson 1994 and Borella 2000). The n-trial K-S test in this dissertation need not delete any empirical data and can offer approximate analytical distribution.

An applicable and more convincing test method of determining analytical distribution with deviation estimation is required in processing large and variable network traffic.

4.2.2.2. INTER-ARRIVAL TIME

It is ideal to keep a stable game packet inter-arrival time. Different types of traffic often had different impacts on the game traffic. The impacts from different traffic on game traffic packet inter-arrival time need to be further explored. If there is relation between the type of traffic and game inter-packet time, in order to keep good game packet inter-arrival time characteristics, routing technique may consider the type of dominating traffic on network link besides all other factors such as geographical distance, congestion level, and so on.

4.2.2.3. DELAY WITH ROUTING TECHNIQUE

Han (2002) found that delay performance varied noticeably according to queue management techniques.

Game traffic has small packet size. However, the current design of routers does not favor such small packets. Game traffic may get better transfer if the design of routing changes. The less impact on current service of router by introducing new design of router, the more likely the new design can be used. Simulation is a good way to evaluate any new design with the gain on game traffic (short) packets and the loss on longer packets. Ns2 simulator has advantage because it supports different routing policies such as Distance-Vector, Link-State, Protocol Independent Multicast-Sparse Mode.

4.2.2.4. DELAY WITH SERVICE POLICY

Another possible approach to decrease game traffic delay is to make use of different network service policies, for example, Integrated Services (IntServ) and Differentiated Services (DiffServ).

IntServ provides guarantees for end-to-end Quality of Service through explicit reservation of resources. Every application that requires some kind of guarantees has to make an individual reservation. However, all intermediary routers on a path should have IntServ capability and they maintain per-flow state information. IntServ can only work on a small network, because it is difficult to keep track of all of the reservations in a large network. The FPS game players usually select a server with short distance in the server-list of the game server selector, all players are usually in short distance from the game server, the game server-clients network is therefore usually small. Game traffic with IntServ needs further exploring.

A more hopeful help is DiffServ. DiffServ classifies packets into classes, depending on the value of the code-point in the packet's IP header. Every router in DiffServ acts on classes depending on a Per Hop Behaviour, although DiffServ does not provide absolute guarantees. Ke (2005) and Shieh (2001) used DiffServ to improve the performance of video streaming by decreasing delay of critical packets in streaming,

Carrig et al (2005) used DiffServ to improve game traffic delay without heavily impact to other traffic's transfer. It can be further explored using DiffServ to improve game traffic delay. A possible starting step is to classify packets into two classes, the class of delay-sensitive and the class of non-delay-sensitive. Game packets belong to the delay-sensitive class, some other applications belong to the non-delay-sensitive class. A new scheduler will be designed, in order to get quicker game traffic transfer in busy network links. Simulation will be performed to check the effect of new scheduler. Ns2 simulator has advantage because it supports different service policies including IntServ and DiffServ.

4.2.2.5. OTHER WORK

Some other possible future work is as follows:

A traffic trace database of the game Quake can be built, the database will be open to access on the Internet.

Game traffic is generated by player actions, and game traffic behaviour inversely affects game performance. Player action, game traffic, and game performance are tightly correlated. The connections among them can be further explored. For example, because the metric behaviour of packet inter-arrival time was successfully explored in both previous research and this research, and the relation between game performance and inter-arrival time was further discussed in this dissertation, possible further research can select one other metric: packet loss. The impact on game performance from packet loss can be explored.

Different games have different network traffic, but they have some similar characteristics. Traffic characteristics such as small packet size and short inter-send time were found in both previous research and this research. Newer and better FPS games are in the game world, the method of exploring game traffic behaviour in this dissertation can be used to other FPS games. The benchmark of game traffic evaluation can likely be obtained after exploring traffic of several games.

The method of exploring traffic behaviour in this dissertation can be used to some other Internet applications with minor modification.

REFERENCES

- Alberto Dainotti, Antonio Pescapè, Giorgio Ventre, et al. 2005. A Packet-level Traffic Model of Starcraft. HOT-P2P '05 Second International Workshop on Hot Topics in Peer-to-Peer Systems.
- Armitage G.J., 2003. An Experimental Estimation of Latency Sensitivity In Multiplayer Quake 3, Technical Report.
- Bangun, R.A. and Dutkiewicz, E. 2000. Modelling Multi-player Games Traffic. Proceedings International Conference on Information Technology: Coding and Computing.
- Borella M.S., 2000. Source Models of Network Game Traffic. Computer Communications, vol 23(4), pp. 403-410.
- Bussiere J.A., Zander S, 2006. Enemy Territory Traffic Analysis, Technical Report, Swinburne University of Technology, Melbourne, Australia, Counter-Strike game software: Retrieved from www.counter-strike.com
- Carrig B., Denieffe D. and Murphy J., 2005. Delivering Quality of Service for Gaming Applications in Best Effort Networks, <http://csiweb.ucd.ie/Staff/jmurphy/publications/1131.pdf#search=%22game%20network%20traffic%20packet%20Quake%20%22>
- Cesar H.D. et al., 2002. Modeling Action and Strategy Internet-Games Traffic, Vehicular Technology Conference.
- Counter-Strike, referenced in 2007. www.counter-strike.net
- Erramilli A., Narayan O. and Willinger W., 1996. Experimental Queuing Analysis with Long-Range Dependent Packet Traffic. IEEE/ACM Trans. Networking, 4(2), pp. 209-223.
- Construction Of Exact Two-Sided Confidence Intervals Based On The Binomial Distribution. Engineering Statistics Handbook by Information Technology Laboratory, Retrieved 2006 from www.itl.nist.gov/div898/handbook/prc/section2/prc241.htm
- Färber J., 2002. Network Game Traffic Modelling, 1st Workshop on Network and System Support for Games, pp. 53-57
- Färber J., 2002. Traffic Modelling for Fast Action Network Games, Special edition 2003 of journal Multimedia Tools and Applications
- Feldmann A., Gilbert A.C., Huang P., Willinger W., 1999. Dynamics of IP traffic: A Study of the Role of Variability and the Impact of Control. Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication Table of Contents.
- Feng W.C., Chang F., Feng W.C, Walpole O., 2002. Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server, Proceedings of Internet Measurement Workshop

- Feng W.C., Chang F., Feng W.C, Walpole O., 2005. A traffic characterization of popular on-line games. *IEEE/ACM Transactions on Networking (TON)*. 13(3), pp. 488 - 500.
- Gargolinski S., Christopher P. and Claypool M., 2005. Game Server Selection for Multiple Players, *NetGames'05*
- Gnuplot 2004. <http://www.gnuplot.info>
- Jehaes T. et al., 2003. Access Network Delay in Networked Games, *Proceedings of the 2nd Workshop on Network and System Support for Games*.
- Joyce S. 2000. Traffic on Internet. Research Report, University of Waikato, Retrived 2002 from <http://wand.cs.waikato.ac.nz/wand/publications/sarah-420.pdf>
- Ke C.H., Shieh C.K., Hwang W.S., Ziviani A., 2005. A Two Markers System for Improved MPEG Video Delivery in a DiffServ Network. *IEEE Communications Letters*, IEEE Press, ISSN: 1089-7798, 9(4), pp. 381-383.
- Han J.S. et al., 2002. Study Of Delay Patterns Of Weighted Voice Traffic Of End-To-End Users on the VoIP Network, *International Journal of Network Management*, 12(5), pp. 271-280.
- Kim J., Choi J., 2005. Traffic Characteristics of a Massively Multi-player Online Role Playing Game, *Proceedings of 4th ACM SIGCOMM Workshop on Network and System Support for Games*.
- Lang T., and Armitage G.J., 2003. A Ns2 Model for the Xbox System Link Game Halo. *Australian Telecommunications Networks & Applications Conference 2003 (ATNAC 2003)*.
- Lang T., Branch P., Armitage G., 2004. A Synthetic Traffic Model for Quake 3, *ACE'04*.
- LaPoint D., Claypool M., Winslow J., 2003. Network Analysis of Counter-strike and Starcraft. *Proceedings of the 22nd IEEE International Performance, Computing, and Communications Conference (IPCCC)*.
- Legout A. and Biersack E.W., 2000. PLM: Fast Convergence for Cumulative Layered Multicast Transmission Schemes, *Special issue on proceedings of ACM SIGMETRICS*.
- McCreary S. and Claffy K., 2000. Trends in Wide Area IP Traffic Patterns: A View from Ames Internet Exchange. *Proceedings of ITC Specialist Seminar on Measurement and Modeling of IP Traffic*.
- Nichols J., Claypool M., 2004. The effect of Lantency on Online Madden NFL Football. *NOSSDAV'04*.
- Park K., Kim G. and Crovella M.E., 1996. On the Relationship Between File Sizes Transport Protocols, and Self-Similar Network Traffic. *Int'l Conf. Network Protocols*, IEEE CS Press, pp.171-180.
- Pavlicic A.M., Armitage G, 2003. Quake 3 Packet Inter-Arrival and Length Over the Internet. Technical Réport, from <http://caia.swin.edu.au/reports/030919B/>
- Paxson V., 1994. Empirically-Derived Analytical Models of Wide-Area TCP Connections. *IEEE/ACM Transactions on Networking*, 2(4), pp. 316-336.

- Paxson V., 1996. An Introduction to Internet Measurement and Modeling. Tutorial given at ACM SIGCOMM '96.
- Pederson S. P. and Johnson M. E., 1990. Estimating Model Discrepancy. *Technometrics*, 32(3), pp. 305-314.
- QuakeWorld, Retrieved 2005 from <http://wiki.quaksrc.org/index.php/FTE%20QuakeWorld>
- Quake III Arena, Retrieved 2005 from http://en.wikipedia.org/wiki/Quake_III_Arena
- Sarvotham S. et.al., 2001. Connection-level Analysis and Modeling of Network Traffic. ACM SIGCOMM Internet Measurement Workshop.
- Sharafeddine S. et al., 2003. On Traffic Characteristics and Bandwidth Requirements of Voice over IP Applications. Eighth IEEE International Symposium on Computers and Communications.
- Shieh C.K., Ke C.H, Hwang W.S., Chiu J.F., 2001. A Transparent Packet Pre-marking Method for DiffServ. IEEE International Conference on Communications, v(9), p2897-2900.
- Zhou Q., Astheimer P., Miller C., 2004. Analysis and Modeling of Multiuser Realtime Internet Applications. Proceedings of Industrial Simulation Conference 2004, Spain.
- Zhou Q., Miller C., Bassilious V., 2008. First Person Shooter Multiplayer Game Traffic Analysis. Proceedings of the Eleventh IEEE International Symposium on Object, Component, and Service-Oriented Real-Time Distributed Computing, USA.

APPENDIX A Calculation of M-Trial Test

The number of failure in a 30-trial test is expected to be less than or equal to 5 in 10% confidence interval.

If the confidence interval is 10%, it is based on the central 90% confidence interval (5% in both sides). The 90% confidence interval indicates defective (failure) rate is 10%, therefore defective rate p is 10% (0.1).

According to the “Engineering Statistics Handbook” by Information Technology Laboratory (Binomial Distribution 2006), the defective rate in a sample follows the binomial distribution, when sample-size is small (if $n < 50$).

Based on this, if a test is performed 30 times and the confidence interval is 10%, it is the binomial distribution with $n = 30$ and $p = 0.1$. The results of calculation of the binomial distribution are shown in Table 18. The 90% confidence interval is indicated by the shaded cells. These are determined by having a cumulative probability between 5% and 95%. Therefore the number of failure in a 30-trial test is expected to be between 1 and 5 (inclusive) on 10% confidence interval.

Number of failures	Probability of n failures (%)	Cumulative prob. of n failures (%)
0	4.2391	4.2391
1	14.1304	18.3695
2	22.7656	41.1351
3	23.6088	64.7439
4	17.7066	82.4505
5	10.2305	92.6810
6	4.7363	97.4173
7	1.8043	99.2216
8	0.5764	99.7980
9	0.1565	99.9545
10	0.0365	99.9910
11	0.0074	99.9984
12	0.0013	99.9997
13	0.0002	99.9999
...
...
30	0	100

Table 23 Calculation of binomial distribution

The above example only offers the number of failure in a 30-trial test is expected to be between 1 and 5, but 0 is not included. Why does the calculation of the binomial

distribution not support zero-number failure in a 30-trial test? It is because the confidence interval level is not in very strict level. If stricter confidence interval level is used, for example, 5%, the calculation of the binomial distribution will support zero-number failure in a 30-trial test.

Therefore the number of failure in a 30-trial test is expected to be less than or equal to 5 in the confidence interval of no less than 90% (90% level is sometimes represented as 10% level).

APPENDIX B Pareto Distribution

The distribution with probability density function and distribution function

$$P(x) = \frac{\alpha b^\alpha}{x^{\alpha+1}}$$

$$D(x) = 1 - \left(\frac{b}{x}\right)^\alpha$$

defined over the interval $x \geq b$.

The n th raw moment is

$$\mu'_n = \frac{\alpha b^n}{\alpha - n}$$

The n th central moment is

$$\begin{aligned} \mu_n &= \alpha b^n \Gamma(\alpha - n) {}_2\tilde{F}_1\left(\alpha - n, -n; 1 + \alpha - n; \frac{\alpha}{\alpha - 1}\right) \\ &= (1 - \alpha)^{\alpha - n} (-\alpha)^{n - \alpha} \alpha b^n B\left(\frac{\alpha}{\alpha - 1}; \alpha - n, n + 1\right), \end{aligned}$$

for $\alpha > n$ and where $\Gamma(z)$ is a gamma function, ${}_2\tilde{F}_1(\alpha, b; c; z)$ is a regularized hypergeometric function, and $B(z; \alpha, b)$ is a beta function.

APPENDIX C Composition of Internet Delay

The end-to-end delay of network is the time required for a packet to traverse the network from one endpoint to the other. The Internet end-to-end delay is composed of propagation delay, transmission delay and queuing delay.

PROPAGATION DELAY

Propagation delay is the time the physical signal traverses the path. It cannot be significantly decreased due to fundamental physical reasons. The speed of light in typical glass fibre is about 200 000 km/sec. For example, the propagation delay to send a signal to a computer on the other end of the earth and get a response back is 200 ms (the circumference of earth is about 40 000 km).

TRANSMISSION DELAY

Transmission delay is further divided into serialization delay, packet processing delay, and modem delay.

Serialization delay: Serialization delay is the time of placing data onto a circuit or network wire. It is the value of dividing the packet size by the capacity of the link. That is, per-byte transmission time multiplied by total bytes. In high-speed network connections, serialization delay is very small and negligible compared to the delay in glass fibre.

Packet processing delay: Packet processing delay consists of all delays needed to process the packet in network nodes. This includes route look-up delay, delay due to the Forward Error Correction process (for example, the interleaving delay of ADSL), etc.

Modem delay: Users required to dial in to an Internet Service Provider using a modem will add significant delay compared to Asymmetric Digital Subscriber Line and cable access. The data rate of modem on serial port is low compared to the backbone Internet (and modem usually has an extra waiting time of 50 ms).

QUEUING DELAY

Queuing delay is the time a packet spends in router queues (packets are buffered in router and wait for transfer) and it is variable. For an unloaded network it is negligible, for heavily congested network it is the main delay component.

If a router does not have enough buffers to store all incoming packets, packets may be dropped and packet loss can cause delay. Some protocols will wait for some period to receive a missing packet before they initiate recovery. Many lost packets will increase the end-to-end delay at the application level. For example, FTP will reduce its rate of transmission when it detects a loss, and that can cause further application delay.

The published papers (pages 108-118 of the thesis) cited below have been removed from the e-thesis due to copyright restrictions:

Zhou, Q., Astheimer, P. and Miller, C.J. (2004) Analysis and modelling of multiuser realtime Internet applications. In: Industrial simulation - International Conference, Malaga, Spain, 2004. Ghent: Eurosis, pp.405-409.

Zhou, Q., Miller, C.J. and Bassilious, V. (2008) First person shooter multiplayer game traffic analysis. In: 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), Orlando, 5-7 May 2008. DOI: 10.1109/ISORC.2008.28